

# cs225f20-b.sg

[Dashboard](#) / [My courses](#) / [cs225f20-b.sg](#) / [21 September - 27 September](#) / [Assignment 2 upload](#)

- Description
- Submission view

## Grade

Reviewed on Thursday, 24 September 2020, 7:11 PM by Automatic grade  
**grade:** A

**Assessment report**[\[-\]](#)  
[\[+\]](#)**Summary of tests**

Submitted on Thursday, 24 September 2020, 7:11 PM ([Download](#))

### IFormatter.h

```
1 #pragma once
2 #include <string>
3
4 class IFormatter
5 {
6 public:
7     virtual IFormatter& insertHeader1(std::string s) = 0;
8     virtual IFormatter& paragraph(std::string s) = 0;
9     virtual IFormatter& insertHeader2(std::string s) = 0;
10    virtual IFormatter& blockquote(std::string s)= 0;
11    virtual ~IFormatter() = default;
12};
```

### Text.cpp

```
1 #include "Text.h"
2 #include <iostream>
3
4 Text::Text(std::ostream& stream) :os{ stream } {}
5
6 IFormatter& Text::insertHeader1(std::string s)
7 {
8     os << s << std::endl;
9     return *this;
10 }
11
12 IFormatter& Text::paragraph(std::string s)
13 {
14     static int count = 0;
15     ++count;
16
17     os << s << std::endl;
18
19     if (count <= 2)
20     {
21         std::cout << std::endl;
22     }
23
24     return *this;
25 }
26
27 IFormatter& Text::insertHeader2(std::string s)
28 {
29     os << s << std::endl;
30     return *this;
31 }
32
33 IFormatter& Text::blockquote(std::string s)
34 {
35     os << "\"" << s << "\"" << std::endl;
36     return *this;
37 }
38
39 IFormatter* construct(std::ostream& stream)
40 {
41     IFormatter* result = new Text(stream);
42     return result;
43 }
44
45 Text::~Text(){}

```

### Html.cpp

```
1 #include "Html.h"
2
3 Html::Html(std::ostream& stream) :os{ stream } {}
4
5 Formatter& Html::insertHeader1(std::string s)
6 {
7     std::cout << "<!doctype html>\n"
8               << "<html lang=en>\n"
9               << "<style>blockquote { border-left: 4px solid #d0d0d0;"
10              << " padding: 4px; }</style>\n"
11              << "<head>\n"
12              << "\t" << "<meta charset=utf-8>\n"
13              << "\t" << "<title>Result</title>\n"
14              << "</head>\n"
15              << "<body>\n";
16
17     std::cout << "\t" << "<h1>" << "1. " << s << "</h1>" << std::endl;
18     return *this;
19 }
20
21 Formatter& Html::paragraph(std::string s)
22 {
23     std::cout << "\t" << "<p>" << s << "</p>" << std::endl;
24     return *this;
25 }
26
27 Formatter& Html::insertHeader2(std::string s)
28 {
29     static int count = 0;
30     ++count;
31
32     std::string paragraph;
33
34     if (count == 1)
35     {
36         paragraph = "1.1. ";
37     }
38     else
39     {
40         paragraph = "1.2. ";
41     }
42
43     os << "\t" << "<h2>" << paragraph << s << "</h2>" << std::endl;
44     return *this;
45 }
46
47 Formatter& Html::blockquote(std::string s)
48 {
49     std::cout << "\t" << "<blockquote>" << s << "</blockquote>" << std::endl;
50     return *this;
51 }
52
53 Formatter* construct(std::ostream& stream)
54 {
55     Formatter* result = new Html(stream);
56     return result;
57 }
58
59 Html::~Html()
60 {
61     std::cout << "</body>" << std::endl;
62     std::cout << "</html>" << std::endl;
63 }
```

Markdown.cpp

```
1 #include "Markdown.h"
2 #include <iostream>
3
4 Markdown::Markdown(std::ostream& stream) :os{ stream } {}
5
6 IFormatter& Markdown::insertHeader1(std::string s)
7 {
8     std::cout << std::endl;
9     os << "# 1. " << s << std::endl;
10
11     return *this;
12 }
13
14 IFormatter& Markdown::paragraph(std::string s)
15 {
16     static int count = 0;
17     ++count;
18
19     os << s << std::endl;
20
21     if (count <= 2)
22     {
23         std::cout << std::endl;
24     }
25
26     return *this;
27 }
28
29 IFormatter& Markdown::insertHeader2(std::string s)
30 {
31     static int count = 0;
32     ++count;
33
34     std::string paragraph;
35
36     if (count == 1)
37     {
38         paragraph = "## 1.1. ";
39     }
40     else
41     {
42         paragraph = "## 1.2. ";
43     }
44
45     os << paragraph << s << std::endl;
46
47     return *this;
48 }
49
50 IFormatter& Markdown::blockquote(std::string s)
51 {
52     os << "> " << s << std::endl;
53     std::cout << std::endl;
54
55     return *this;
56 }
57
58 IFormatter* construct(std::ostream& stream)
59 {
60     IFormatter* result = new Markdown(stream);
61     return result;
62 }
63
64 Markdown::~Markdown() {}
```

Text.h

```
1 #pragma once
2 #include "IFormatter.h"
3
4 class Text : public IFormatter
5 {
6 public:
7     std::ostream& os;
8     Text(std::ostream& stream);
9     IFormatter& insertHeader1(std::string s);
10    IFormatter& paragraph(std::string s);
11    IFormatter& insertHeader2(std::string s);
12    IFormatter& blockquote(std::string s);
13    virtual ~Text();
14};
```

Markdown.h

```
1 #pragma once
2 #include "IFormatter.h"
3
4 class Markdown : public IFormatter
5 {
6 public:
7     std::ostream& os;
8     Markdown(std::ostream& stream);
9     IFormatter& insertHeader1(std::string s);
10    IFormatter& paragraph(std::string s);
11    IFormatter& insertHeader2(std::string s);
12    IFormatter& blockquote(std::string s);
13    virtual ~Markdown();
14};
```

Html.h

```
1 #pragma once
2 #include "IFormatter.h"
3 #include <iostream>
4
5 class Html : public IFormatter
6 {
7 public:
8     std::ostream& os;
9     Html(std::ostream& stream);
10    IFormatter& insertHeader1(std::string s);
11    IFormatter& paragraph(std::string s);
12    IFormatter& insertHeader2(std::string s);
13    IFormatter& blockquote(std::string s);
14    virtual ~Html();
15 };
```

[VPL](#)

◀ Assignment 2 files (expected output included)

Jump to...

⌵

Attendance cs225f20-b.sg Tuesday 29/9/2020 1:30pm-3:10pm ▶

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