

cs225f20-b.sg

[Dashboard](#) / [My courses](#) / [cs225f20-b.sg](#) / [5 October - 11 October](#) / [Assignment 3 upload](#)

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

Grade

Reviewed on Tuesday, 13 October 2020, 9:20 PM by Automatic grade
grade: A

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

+-----+
| 71 tests run/71 tests passed |
+-----+

Submitted on Tuesday, 13 October 2020, 9:20 PM ([Download](#))

Constructor.cpp

```
1  #include "Constructor.h"
2  #include "Text.h"
3  #include "Html.h"
4  #include "Markdown.h"
5
6  std::vector<std::string> get_options()
7  {
8      std::vector < std::string > v;
9
10     v.push_back("Text");
11     v.push_back("Markdown");
12     v.push_back("HTML");
13
14     return v;
15 }
16
17 IFormatterFactory* construct(int selected_option)
18 {
19     IFormatterFactory* result = NULL;
20
21     switch (selected_option)
22     {
23         case 0 : result = new Text(); break;
24         case 1 : result = new Markdown(); break;
25         case 2: result = new Html(); break;
26         default: break;
27     }
28
29     return result;
30 }
31
32 void save(ISavable& savable, std::ostream& stream)
33 {
34     savable.save(stream);
35 }
36
```

Html.h

```

1  #ifndef _HTML_H_
2  #define _HTML_H_
3
4  #include "IFormatterFactory.h"
5  #include "ISavable.h"
6
7  class Html : public IFormatterFactory, public IFormatterElement, public ISavable
8  {
9  public:
10     std::string str;
11
12     Html() = default;
13     Html(std::string s): str{s}{};
14
15     IFormatterElement* create_begin() override
16     {
17         std::string newString;
18
19         newString += "<!doctype html>\n";
20         newString += "<html lang=en>\n";
21         newString += "<style>blockquote { border-left: 4px solid #d0d0d0;";
22         newString += " padding: 4px; }</style>\n";
23         newString += "<head>\n";
24         newString += "\t<meta charset=utf-8>\n";
25         newString += "\t<title>Result</title>\n";
26         newString += "</head>\n";
27         newString += "<body>\n";
28
29         return new Html(newString);
30     }
31
32     IFormatterElement* create_header1(std::string s) override
33     {
34         std::string newString;
35
36         newString += "\t<h1>1. ";
37         newString += s;
38         newString += "</h1>\n";
39
40         return new Html{newString};
41     }
42
43     IFormatterElement* create_paragraph(std::string s) override
44     {
45         std::string newString;
46
47         newString += "\t<p>";
48         newString += s;
49         newString += "</p>\n";
50
51         return new Html{newString};
52     }
53
54     IFormatterElement* create_header2(std::string s) override
55     {
56         std::string newString;
57
58         static int count = 0;
59         ++count;
60
61         newString += "\t<h2>";
62
63         if (count == 1)
64         {
65             newString += "1.1. ";
66         }
67         else
68         {
69             newString += "1.2. ";
70         }
71
72         newString += s;
73         newString += "</h2>\n";
74
75         return new Html{newString};
76     }
77
78     IFormatterElement* create_blockquote(std::string s) override
79     {
80         std::string newString;
81
82         newString += "\t<blockquote>";
83         newString += s;
84         newString += "</blockquote>\n";
85
86         return new Html{newString};
87     }
88
89     IFormatterElement* create_end() override
90     {
91         std::string newString;
92
93         newString += "</body>\n";
94         newString += "</html>\n";
95
96         return new Html{newString};
97     }
98
99     void save(std::ostream& stream)
100    {
101        stream << str;
102    }
103
104    ~Html(){}
105 };
106
107 #endif // _HTML_H_

```

Text.h

```
1 #ifndef _TEXT_H_
2 #define _TEXT_H_
3
4 #include "IFormatterFactory.h"
5 #include "ISavable.h"
6
7 class Text : public IFormatterFactory, public IFormatterElement, public ISavable
8 {
9 public:
10     std::string str;
11
12     Text() = default;
13     Text(std::string s): str{s}{};
14
15     IFormatterElement* create_begin() override
16     {
17         std::string newString;
18         return new Text(newString);
19     }
20
21     IFormatterElement* create_header1(std::string s) override
22     {
23         std::string newString;
24
25         newString += s;
26         newString += "\n";
27
28         return new Text(newString);
29     }
30
31     IFormatterElement* create_paragraph(std::string s) override
32     {
33         std::string newString;
34
35         newString += s;
36         newString += "\n";
37
38         static int count = 0;
39         ++count;
40
41         if(count<=2)
42             newString += "\n";
43
44         return new Text(newString);
45     }
46
47     IFormatterElement* create_header2(std::string s) override
48     {
49         std::string newString;
50
51         newString += s;
52         newString += "\n";
53
54         return new Text(newString);
55     }
56
57     IFormatterElement* create_blockquote(std::string s) override
58     {
59         std::string newString;
60
61         newString += "\"";
62         newString += s;
63         newString += "\"";
64         newString += "\n";
65
66         return new Text(newString);
67     }
68
69     IFormatterElement* create_end() override
70     {
71         std::string newString;
72         return new Text(newString);
73     }
74
75     void save(std::ostream& stream)
76     {
77         stream << str;
78     }
79
80     ~Text() {}
81 };
82
83 #endif // TEXT H
```

Markdown.h

```

1  #ifndef _MARKDOWN_H_
2  #define _MARKDOWN_H_
3
4  #include "IFormatterFactory.h"
5  #include "ISavable.h"
6
7  class Markdown : public IFormatterFactory, public IFormatterElement
8  , public ISavable
9  {
10 public:
11     std::string str;
12
13     Markdown() = default;
14     Markdown(std::string s): str{s}{};
15
16     IFormatterElement* create_begin()
17     {
18         std::string newString;
19
20         newString += "\n";
21         return new Markdown(newString);
22     }
23
24     IFormatterElement* create_header1(std::string s) override
25     {
26         std::string newString;
27
28         newString += "# 1. ";
29         newString += s;
30         newString += "\n";
31
32         return new Markdown(newString);
33     }
34
35     IFormatterElement* create_paragraph(std::string s) override
36     {
37         std::string newString;
38
39         newString += s;
40         newString += "\n";
41
42         static int count = 0;
43         ++count;
44
45         if (count <= 2)
46         {
47             newString+= "\n";
48         }
49
50         return new Markdown(newString);
51     }
52
53     IFormatterElement* create_header2(std::string s) override
54     {
55         std::string newString;
56
57         static int counter = 0;
58         ++counter;
59
60         if (counter == 1)
61         {
62             newString += "## 1.1. ";
63         }
64         else
65         {
66             newString += "## 1.2. ";
67         }
68
69         newString += s;
70         newString += "\n";
71
72         return new Markdown(newString);
73     }
74
75     IFormatterElement* create_blockquote(std::string s) override
76     {
77         std::string newString;
78
79         newString += "> ";
80         newString += s;
81         newString += "\n\n";
82
83         return new Markdown(newString);
84     }
85
86     IFormatterElement* create_end() override
87     {
88         std::string newString;
89         return new Markdown(newString);
90     }
91
92     void save(std::ostream& stream)
93     {
94         stream << str;
95     }
96
97     ~Markdown() {}
98 };
99
100 #endif // _MARKDOWN_H_
101

```

IFormatterFactory.h

```
1 #ifndef _IFORMATTERFACTORY_H_
2 #define _IFORMATTERFACTORY_H_
3
4 #include <string>
5 #include "IFormatterElement.h"
6
7 class IFormatterFactory
8 {
9 public:
10     virtual IFormatterElement* create_begin() = 0;
11     virtual IFormatterElement* create_header1(std::string s) = 0;
12     virtual IFormatterElement* create_paragraph(std::string s) = 0;
13     virtual IFormatterElement* create_header2(std::string s) = 0;
14     virtual IFormatterElement* create_blockquote(std::string s) = 0;
15     virtual IFormatterElement* create_end() = 0;
16     virtual ~IFormatterFactory() = default;
17 };
18
19 #endif // IFORMATTERFACTORY H
```

IFormatterElement.h

```
1 #ifndef _IFORMATTERELEMENT_H_
2 #define _IFORMATTERELEMENT_H_
3
4 #include "ISavable.h"
5
6 class IFormatterElement
7 {
8 public:
9     virtual ~IFormatterElement() = default;
10 };
11
12 #endif // IFORMATTERELEMENT H
```

[VPL](#)

◀ Assignment 3 files

Jump to...

Attendance cs225f20-b.sg Tuesday
13/10/2020 1:30pm-3:10pm ▶

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