

Object Oriented Programming - IE

[Home](#) / [My courses](#) / [OOP - IE](#) / [Exam](#) / [Written exam](#)

Quiz navigation




[Finish attempt ...](#)

Time left **1:29:04**

Question 1

Not yet answered

Marked out of 3.00

 Flag question

Define the **HTMLElement**, **HTMLImage**, **HTMLParagraph** and **HTMLBuilder** classes such that the following C++ code is correct, all asserts pass and its results are the ones indicated in the comments. Provide specification for the `getHTMLString()` method.

```
void fct1() {
    HTMLElement* p1 = new HTMLParagraph{ "Examination" };
    assert(p1->getHTMLString() == "<p>Examination</p>");
    HTMLElement* p2 = nullptr;
    HTMLElement* i1 = new HTMLImage{ "a.jpg" };
    HTMLElement* i2 = new HTMLImage{ "b.jpg" };
    assert(i2->getHTMLString() == "<img>b.jpg</img>");
    HTMLBuilder<HTMLElement*> html{};
    try {
        html += p2;
        assert(false);
    } catch (runtime_error& e) {
        assert(strcmp(e.what(), "Cannot add a null element!") == 0);
    }

    ((html += p1) += i1) += i2;
    cout << html; // prints: <html><body><p>Examination</p><img>a.jpg</img><img>b.jpg</img></html></body>
    delete p1; delete p2;
    delete i1; delete i2;
}
```





Question **2**




Not yet
answered


Marked out of
2.00

Flag question

Maximum file size: 2MB, maximum number of files: 1

 Files

Accepted file types

All file types

Specify the function below and write a C++ test function for it.

```
int fct1(int x) {  
    if (x % 10 == x) {  
        throw std::runtime_error{"Number contains a single digit"};  
    }  
    stringstream s;  
    s << x;  
    string str = s.str();  
    int y = str.size();  
    int i = 0;  
    while (i < y - 1 && str[i] <= str[i + 1]) {  
        i++;  
    }  
    return (i == y - 1);  
}
```






Maximum file size: 2MB, maximum number of files: 1





 Files

Accepted file types
All file types

Question 3

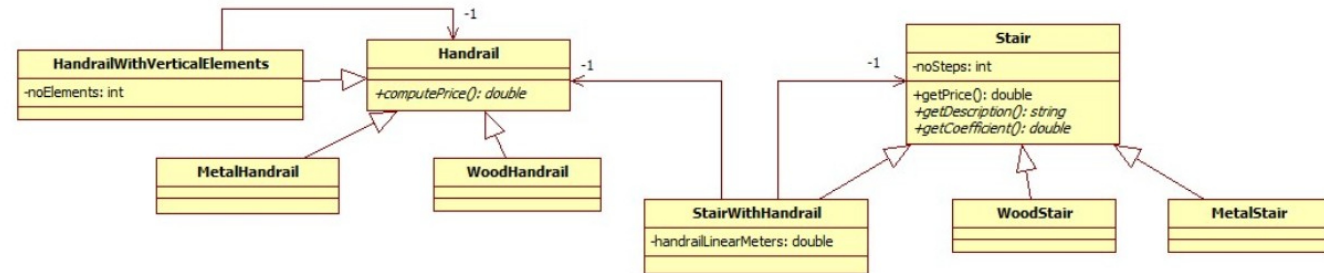
Not yet answered

Marked out of 4.00

Flag question



Write a C++ application for building stairs, as follows:




- The class **Handrail** is abstract, it contains the abstract method `computePrice()`. [0.5p]
- The classes **WoodHandrail** and **MetalHandrail** represent specific types of handrails. The price for one linear meter of wood handrail is 2, while for metal handrails this price is 2.5. [0.5p]
- The price for one linear meter of handrail with vertical elements (**HandrailWithVerticalElements**) is the price of the aggregated handrail, to which the price of elements is added. The price for one element it 5. [0.5p]
- The abstract class **Stair** represents a generic stair. The price of a simple stair is computed as the coefficient multiplied by the stair's number of steps. [0.5p]
- The description for a wood stair (**WoodStair**) is "wood stair", while for a metal one (**MetalStair**), this description is "metal stair". The coefficient for a wood stair is 1.5, while for a metal one it is 2. [0.5p]
- The price of a stair with handrail (**StairWithHandrail**) is the price of the aggregated stair, to which the price of the aggregated handrail multiplied by the number of linear meters is added. The coefficient for a stair with handrail is 1, while the description is the description of the aggregated stair, to which the number of linear meters of handrail is added. [0.5p]
- Create the following objects: a simple wood stair, with 20 steps; a metal stair, with 10 steps, with 5 linear meters of metal handrail; a wood stair with 10 steps, with 5 linear meters of wood handrail, with 10 vertical elements. Print the description and price for each stair. Make sure the memory is correctly managed. [1p]











Maximum file size: 2MB, maximum number of files: 1






 Files





 Files

Accepted file types