



Part 2 - Practice

1 OF 1 QUESTIONS REMAINING

Test Content

Question 1

10 Points

Given the following struct that describes a Gem,

```
1.structGem{
2.string name;
3.string colour;
4.int shine;
5.bool polished;
6.};
```

Create a Box class has the following properties:

- An STL container of Gems as a data member
- A += operator that adds Gems to the box
- A -= operator that removes Gems from the Box based on a given string (name) to search for. If the string is found then remove any Gems that match. If there isn't a match at all then throw an exception of string type with a relevant error message
- A function called process that doesn't receive any parameters nor returns any value. This function will iterate through the Gems in the Box and polish them if they aren't yet polished. Polishing a Gem will cause its shine value to increase by 20. For this functionality incorporate the use of threads to divide the work.
- A display function that displays the contents of the box

The following main will use the Box class:

```
1.int main(){
2.Box b;
3.
4.  cout <<"\nAdding Gems"<< endl;
5.  b +=Gem{"Emerald","Green",20,false};
6.  b +=Gem{"Ruby","Red",50,true};
7.  b +=Gem{"Sapphire","Blue",80,false};
8.  b +=Gem{"Onyx","Black",60,false};
9.  b +=Gem{"Opal","White",10,false};
10.  b.display();
11.
12.  cout <<"\nRemoving Gems"<< endl;
13.try{
14.    b -= "Ruby";
15.    b -= "Emerald";
16.    b -= "Diamond";
17.    b.display();
18.}
19.catch(string c){
20.    cout << c << endl;
21.}
22.
23.  cout <<"\nPolishing Gems"<< endl;
24.  b.process();
25.  b.display();
26.
27.
28.  cout <<"\nVector of Boxes"<< endl;
29.  vector<Box*> boxes;
30.  boxes.push_back(newBox());
31.  boxes.push_back(newBox());
32.  boxes.push_back(newBox());
33.
34.for(auto x : boxes){
35.    x->operator+=(Gem{"Emerald","Green",20,false});
36.    x->operator+=(Gem{"Ruby","Red",90,false});
37.    x->operator+=(Gem{"Sapphire","Blue",40,false});
38.}
39.
40.for(auto x : boxes){
41.delete x;
42.}
43.}
```

As an extra task, modify the Box class to instead have a container of Gem pointers and apply the use of smart pointers to the solution.

Use the editor to format your answer

Questions Filter (1) ▼

Save and Close

Submit