Part 4 - Practice

1 OF 1 QUESTIONS REMAINING

Test Content

Question 1

10 Points

There are 6 errors in the following code (eg. Compile, Undefined Behavior, Memory Leak, Runtime Crash...).
Identify them in the following format:

File, Line Number
Type Of Error
Description of Error
Fix for Error

Make use of the following hints to locate each error:

1. Error #1 located in sample.h/main.cpp related to the use of redefinitions
2. Error #2 located in sample.h/main.cpp related to the use of objects
3. Error #3 located in sample.cpp/main.cpp related to the use of types
4. Error #4 located in main.cpp related to order of precedence
5. Error #5 located in main.cpp related to threading
6. Error #6 located in sample.cpp/main.cpp related to the use of shared states

```
1.// sample.h
2.
3.#include<iostream>
4.#include<string>
5.#include<vector>
6.#include<future>
7.
8.structGem{
9.  std::string name{};
10.unsigned weight{};
11.};
12.
13.classTreasureBox{
14.  std::vector<Gem> gems;
15.unsigned weight{};
16.public:
17.TreasureBox(const std::string* names,unsigned* weights,unsigned size);
18.TreasureBox&operator+=(constGem& g);
19.void update(const std::string& name);
20.  std::ostream& display()const;
21.};
22.
23.void task(std::promise<unsigned>& p);
24.
```

```
1.// sample.cpp
2.#include<iostream>
3.#include<string>
4.#include<vector>
5.#include<algorithm>
6.#include<numeric>
7.#include<future>
8.#include"sample.h"
9.
10.TreasureBox::TreasureBox(const std::string* names,unsigned* weights,unsigned size){
11.for(unsigned x =0; x < size;++x){
12.    gems.push_back(Gem{ names[x], weights[x]});
13.    weight += weights[x];
14.}
15.}
16.
17.TreasureBox&TreasureBox::operator+=(constGem& g){
18.  gems.push_back(g);
19.  weight += g.weight;
20.return*this;
```

```
21.}
22.
23.voidTreasureBox::update(const std::string& name){
24.  std::vector<Gem>::iterator itr;
25.for(itr = gems.begin(); itr != gems.end();++itr){
26.if(itr->name == name){
27.unsigned sum =0;
28.      sum = std::accumulate(name.begin(), name.end(),0);
29.      itr->weight += sum;
30.}
31.}
32.}
33.
34.std::ostream&TreasureBox::display()const{
35.  std::cout <<"TreasureBox Contents"<< std::endl;
36.for(auto& g : gems){
37.    std::cout << g.name <<" : "<< g.weight << std::endl;
38.}
39.return std::cout;
40.}
41.
42.void task(std::promise<unsigned>& p){
43.Gem gems[]{{"Red",10},{"Green",20},{"Blue",30}};
44.
45.auto idx = std::find_if(gems, gems +2,[](constGem& g){
46.return g.name =="Green";
47.});
48.}
```

```
1.// main.cpp
2.#include<iostream>
3.#include<string>
4.#include<string_view>
5.#include<vector>
6.#include<algorithm>
7.#include<numeric>
8.#include<future>
9.#include<thread>
10.#include"sample.h"
11.#include"sample.h"
12.
13.int main(){
14.
15.  std::string strs[]{"Ruby","Emerald","Sapphire","Opal"};
16.unsigned nums[]{10,20,30,40};
```

```
17.
18.  std::unique_ptr<TreasureBox> tbr(newTreasureBox());
19.  std::unique_ptr<TreasureBox> ptr(newTreasureBox(strs, nums,4));
20.
21.  std::string_view sv ="Ruby";
22.  std::string s ="Opal";
23.  ptr->update(sv);
24.  ptr->update(s);
25.  ptr->display();
26.
27.*ptr +=Gem{"Diamond",50}+=Gem{"Amber",60};
28.(*ptr +=Gem{"Pearl",70}).display();
29.
30.  std::promise<unsigned> pro;
31.  std::future<unsigned> fut = pro.get_future();
32.
33.  std::thread t1;
34.  t1.join();
35.  t1 = std::thread(task, std::ref(pro));
36.  std::cout <<"Promised Future: "<< fut.get()<< std::endl;
37.  t1.join();
38.}
```

Use the editor to format your answer

Questions Filter (1) ▼

Save and Close

Submit