Extra Debug Practice

10 OF 10 QUESTIONS REMAINING

Test Content

Question 1

1 Point

The code below contains **three** (3) syntactic errors under C++17 standard (errors that are caught by a compiler or generate crashes/undefined behaviour at runtime). Your task is to identify each one by the file name and line number and **explain why the error appears, what C++ standard rule is broken, what C++ feature is misused and how the error should be fixed**. Your answer will be evaluated based on **clarity of the text** and the show of **understanding of the concepts** that are involved in the error.

```
// assign.h
#ifndef SDDS_ASSIGN_H
#define SDDS_ASSIGN_H
#include <iostream>
#include <string>
#include <vector>

class Assign
{
    std::vector<long long> m_data;
    std::string  m_title{ "" };
    double m_value{ -1 };
public:
    Assign(std::string str, double value);

    std::ostream& log() const;

    Assign& update();

    double getData(size_t step) const;
};
#endif
```

```
// assign.cpp
#include <iostream>
#include <string>
#include <utility>
#include "assign.h"

using namespace std;

Assign::Assign(string str, double value)
{
    m_title = str;
    m_value = value;
    for (auto i = 0; i < 123; ++i)
        m_data.push_back(static_cast<long long>(i * value));
}

Assign& Assign::update()
{
    if (m_data.size() == 0)
        catch "An error appeared. Cannot update";
    for (auto it = m_data.begin(); it != m_data.end(); ++it)
        *it += 5;
```

```
    return *this;
}

double Assign::getData(size_t step) const
{
    double sum = 0;
    for (auto i = step; i < m_data.size(); i += step)
        sum += m_data[i];
    return sum;
}

ostream& Assign::log() const
{
    clog << m_title << " : " << m_value << "\n\t";
    for (auto& elem : m_data)
        clog << elem << ", ";

    return clog << '\n';
}
```

```
// main.cpp
#include <iostream>
#include <string>
#include <string_view>
#include <thread>
#include <future>
#include <numeric>
#include "Assign.h"
using namespace std;

thread_local int step = 0;

void task(promise<double> p)
{
    vector<Assign> VE;
    VE.push_back({ "AAAAAAAA", 1.2 });
    VE.push_back({ "BBBBBBBB", 2.3 });
    VE.push_back({ "CCCCCCCC", 3.4 });
    VE.push_back({ "DDDDDDDD", 4.5 });
    double s = accumulate(VE.begin(), VE.end(), 0.0,
        [](const double& x, const Assign& e)
        {
            step = (step + 1) % 5;
            return x + e.getData(step);
        });
```

```
26.     p.set_value(s);
27. }
28.
29. int main()
30. {
31.     unique_ptr<Assign> ptr(new Assign("Hello", 2.3));
32.     ptr->update().log();
33.
34.     delete ptr;
35.
36.     const char* strc  = "CCCCCCCC";
37.     string strb, stra = "AAAAAAAA";
38.     string_view strv  = strc;
39.
40.     promise<double> pp;
41.     future<double> ff = pp.get_future();
42.
43.     strc = "BBBBBBBB";
44.
45.     thread t1(task, std::ref(pp));
46.     cout << "Value = " << ff.get() << endl;
47.
48.     t1.join();
49. }
```

Use the editor to format your answer

Question 2

1 Point

The code below contains **three** (3) syntactic errors under C++17 standard (errors that are caught by a compiler or generate crashes/undefined behaviour at runtime). Your task is to identify each one by the file name and line number and **explain why the error appears, what C++ standard rule is broken, what C++ feature is misused and how the error should be fixed**. Your answer will be evaluated based on **clarity of the text** and the show of **understanding of the concepts** that are involved in the error.

```
// assign.h
#ifndef SDDS_ASSIGN_H
#define SDDS_ASSIGN_H
#include <iostream>
#include <string>
#include <list>

class Assign
{
    std::list<long long> m_data;
    std::string  m_title{ "NoName" };
    double m_value{ -1 };
public:
    Assign(std::string str, double field2);

    std::ostream& log() const;

    Assign& update();

    double getData(size_t step) const;
};
#endif
```

```cpp
// assign.cpp
#include <iostream>
#include <string>
#include <utility>
#include "assign.h"

using namespace std;

Assign::Assign(string str, double value)
{
    m_title = str;
    m_value = value;
    for (auto i = 0; i < 123; ++i)
        m_data.push_back(static_cast<long long>(i * value));
}

Assign& Assign::update()
{
    for (auto it = m_data.begin(); it != m_data.end(); ++it)
        *it += 5;
    return *this;
}

double Assign::getData(size_t step) const
{
    double sum = 0;
    for (auto i = step; i < m_data.size(); i += step)
        sum += m_data[i];
    return sum;
}

ostream& Assign::log() const
{
    clog << m_title << " : " << m_value << "\n\t";
    for (auto& elem : m_data)
        clog << elem << ", ";

    return clog << '\n';
}
```

```cpp
// main.cpp
#include <iostream>
#include <string>
#include <string_view>
#include <thread>
#include <future>
#include <numeric>
#include "Assign.h"
using namespace std;

thread_local int step = 0;

void task(promise<double>& p)
{
    vector<Assign> VE;
    VE.push_back({ "AAAAAAAA", 1.2 });
    VE.push_back({ "BBBBBBBB", 2.3 });
    VE.push_back({ "CCCCCCCC", 3.4 });
    VE.push_back({ "DDDDDDDD", 4.5 });
    double s = accumulate(VE.begin(), VE.end(), 0.0,
        [](const double& x, const Assign& e)
        {
            step = (step + 1) % 5;
            return x + e.getData(step);
        });
    p.set_value(s);
}

int main()
{
    unique_ptr<Assign> a = new Assign("Hello", 2.3);
    a->update().log();

    const char* strc  = "CCCCCCCC";
    string strb, stra = "AAAAAAAA";
    string_view strv  = "VVVVVVVV";

    promise<double> pp;
    future<double> ff = pp.get_future();

    strv = stra;

    thread t1(task, std::ref(pp));
    cout << "Value = " << pp.get() << endl;

```

```
46.     t1.join();
47. }
```

Use the editor to format your answer

Question 3

1 Point

The code below contains **three** (3) syntactic errors under C++17 standard (errors that are caught by a compiler or generate crashes/undefined behaviour at runtime). Your task is to identify each one by the file name and line number and **explain why the error appears, what C++ standard rule is broken, what C++ feature is misused and how the error should be fixed**. Your answer will be evaluated based on **clarity of the text** and the show of **understanding of the concepts** that are involved in the error.

```
 1. // assign.h
 2. #ifndef SDDS_ASSIGN_H
 3. #define SDDS_ASSIGN_H
 4. #include <iostream>
 5. #include <string>
 6. #include <vector>
 7.
 8. class Assign
 9. {
10.     std::vector<long long> m_data;
11.     std::string  m_title{ "NoName" };
12.     double m_value{ -1 };
13. public:
14.     Assign(std::string str, double field2);
15.
16.     std::ostream& log() const;
17.
18.     Assign& update();
19.
20.     double getData(size_t step) const;
21. };
22. #endif
```

```cpp
// assign.cpp
#include <iostream>
#include <string>
#include <utility>
#include "assign.h"

using namespace std;

Assign::Assign(string str, double value)
{
    m_title = str;
    m_value = value;
    for (auto i = 0; i < 123; ++i)
        m_data[i] = static_cast<long long>(i * value);
}

Assign& Assign::update()
{
    for (auto it = m_data.begin(); it != m_data.end(); ++it)
        *it += 5;
    return *this;
}

double Assign::getData(size_t step) const
{
    double sum = 0;
    for (auto i = step; i < m_data.size(); i += step)
        sum += m_data[i];
    return sum;
}

ostream& Assign::log() const
{
    clog << m_title << " : " << m_value << "\n\t";
    for (auto& elem : m_data)
        clog << elem << ", ";

    return clog << '\n';
}
```

```cpp
// main.cpp
#include <iostream>
#include <string>
#include <string_view>
#include <thread>
#include <future>
#include <numeric>
#include "Assign.h"
using namespace std;

thread_local int step = 0;

void task(promise<double>& p)
{
    vector VE;
    VE.push_back({ "AAAAAAAA", 1.2 });
    VE.push_back({ "BBBBBBBB", 2.3 });
    VE.push_back({ "CCCCCCCC", 3.4 });
    VE.push_back({ "DDDDDDDD", 4.5 });
    double s = accumulate(VE.begin(), VE.end(), 0.0,
        [](const double& x, const Assign& e)
        {
            step = (step + 1) % 5;
            return x + e.getData(step);
        });
    p.set_value(s);
}

int main()
{
    Assign a("Hello", 2.3);
    a.update().log();

    const char* strc  = "CCCCCCCC";
    string strb, stra = "AAAAAAAA";
    string_view strv  = "VVVVVVVV";

    promise<double> p;
    future<double> f = p.get_future();

    strv = stra;

    thread t1(task, std::ref(p));
```

```
44.     cout << "Value = " << f.get() << endl;
45. }
```

Use the editor to format your answer

Question 4

1 Point

The code below contains **three** (3) syntactic errors under C++17 standard (errors that are caught by a compiler or generate crashes/undefined behaviour at runtime). Your task is to identify each one by the file name and line number and **explain why the error appears, what C++ standard rule is broken, what C++ feature is misused and how the error should be fixed**. Your answer will be evaluated based on **clarity of the text** and the show of **understanding of the concepts** that are involved in the error.

```
 1. // assign.h
 2. #ifndef SDDS_ASSIGN_H
 3. #define SDDS_ASSIGN_H
 4. #include <iostream>
 5. #include <string>
 6. #include <vector>
 7.
 8. class Assign
 9. {
10.     std::vector<long long&> m_data;
11.     std::string  m_title{ "NoName" };
12.     double m_value{ -1 };
13. public:
14.     Assign(std::string str, double field2);
15.
16.     std::ostream& log() const;
17.
18.     Assign& update();
19.
20.     double getData(size_t step) const;
21. };
22. #endif
```

```cpp
// assign.cpp
#include <iostream>
#include <string>
#include <utility>
#include "assign.h"

using namespace std;

Assign::Assign(string str, double value)
{
    m_title = str;
    m_value = value;
    for (auto i = 0; i < 123; ++i)
        m_data.push_back(dynamic_cast<long long>(i * value));
}

Assign& Assign::update()
{
    for (auto it = m_data.begin(); it != m_data.end(); ++it)
        *it += 5;
    return *this;
}

double Assign::getData(size_t step) const
{
    double sum = 0;
    for (auto i = step; i < m_data.size(); i += step)
        sum += m_data[i];
    return sum;
}

ostream& Assign::log() const
{
    clog << m_title << " : " << m_value << "\n\t";
    for (auto& elem : m_data)
        clog << elem << ", ";

    return clog << '\n';
}
```

```cpp
// main.cpp
#include <iostream>
#include <string>
#include <string_view>
#include <thread>
#include <future>
#include <numeric>
#include "Assign.h"
using namespace std;

thread_local step = 0;

void task(promise<double>& p)
{
    vector<Assign> VE;
    VE.push_back({ "AAAAAAAA", 1.2 });
    VE.push_back({ "BBBBBBBB", 2.3 });
    VE.push_back({ "CCCCCCCC", 3.4 });
    VE.push_back({ "DDDDDDDD", 4.5 });
    double s = accumulate(VE.begin(), VE.end(), 0.0,
        [](const double& x, const Assign& e)
        {
            step = (step + 1) % 5;
            return x + e.getData(step);
        });
    p.set_value(s);
}

int main()
{
    Assign a("Hello", 2.3);
    a.update().log();

    const char* strc  = "CCCCCCCC";
    string strb, stra = "AAAAAAAA";
    string_view strv  = "VVVVVVVV";

    promise<double> p;
    future<double> f = p.get_future();

    strv = stra;

    thread t1(task, std::ref(p));
    cout << "Value = " << f.get() << endl;

```

```
46.     t1.join();
47. }
```

Use the editor to format your answer

Question 5

1 Point

The code below contains **three** (3) syntactic errors under C++17 standard (errors that are caught by a compiler or generate crashes/undefined behaviour at runtime). Your task is to identify each one by the file name and line number and **explain why the error appears, what C++ standard rule is broken, what C++ feature is misused and how the error should be fixed**. Your answer will be evaluated based on **clarity of the text** and the show of **understanding of the concepts** that are involved in the error.

```
 1. // assign.h
 2. #ifndef SDDS_ASSIGN_H
 3. #define SDDS_ASSIGN_H
 4. #include <iostream>
 5. #include <string>
 6. #include <vector>
 7.
 8. class Assign
 9. {
10.     std::vector<long long> m_data;
11.     std::string  m_title{ "NoName" };
12.     double m_value{ -1 };
13. public:
14.     Assign(std::string str, double field2);
15.
16.     std::ostream& log() const;
17.
18.     Assign& update();
19.
20.     double getData(size_t step) const;
21. };
22. #endif
```

```cpp
// assign.cpp
#include <iostream>
#include <string>
#include <utility>
#include "assign.h"

using namespace std;

Assign::Assign(string str, double value)
{
    m_title = str;
    m_value = value;
    for (auto i = 0; i < 123; ++i)
        m_data.push_back(static_cast<long long>(i * value));
}

Assign& Assign::update()
{
    for (auto it = m_data.cbegin(); it != m_data.cend(); ++it)
        *it += 5;
    return *this;
}

double Assign::getData(size_t step) const
{
    double sum = 0;
    for (auto i = step; i < m_data.size(); i += step)
        sum += m_data[i];
    return sum;
}

ostream& Assign::log() const
{
    clog << m_title << " : " << m_value << "\n\t";
    for (auto& elem : m_data)
        clog << elem << ", ";

    return clog << '\n';
}
```

```cpp
// main.cpp
#include <iostream>
#include <string>
#include <string_view>
#include <thread>
#include <future>
#include <numeric>
#include "Assign.h"
using namespace std;

thread_local size_t step = 0;

void task(promise<double>& p)
{
    vector<Assign> VE;
    VE.push_back({ "AAAAAAAA", 1.2 });
    VE.push_back({ "BBBBBBBB", 2.3 });
    VE.push_back({ "CCCCCCCC", 3.4 });
    VE.push_back({ "DDDDDDDD", 4.5 });
    double s = accumulate(VE.begin(), VE.end(), 0.0,
        [](const double& x, const Assign& e)
        {
            step = (step + 1) % 5;
            return x + e.getData(step);
        });
    p.set_value(s);
}

int main()
{
    Assign a("Hello", 2.3);
    a.update().log();

    const char* strc  = "CCCCCCCC";
    string strb, stra = "AAAAAAAA";
    string_view strv  = "VVVVVVVV";

    promise<double> p;
    future<double> f = p.get_future();

    strv += stra;

    thread t1(task, p);
    cout << "Value = " << f.get() << endl;

```

```
46.     t1.join();
47. }
```

Use the editor to format your answer

Question 6

1 Point

The code below contains **two** (2) syntactic errors under C++17 standard (errors that are caught by a compiler or generate crashes/undefined behaviour at runtime). Your task is to identify each one by the file name and line number and **explain why the error appears, what C++ standard rule is broken, what C++ feature is misused and how the error should be fixed**. Your answer will be evaluated based on **clarity of the text** and the show of **understanding of the concepts** that are involved in the error.

```cpp
#include <iostream>
#include <array>
using namespace std;

#define PRINT_HEADER(msg) cout << "****  -> " << msg << " <-  ****\n"

bool divBy3(const int& val)
{
    return val % 3 == 0;
}

bool divBy2(const int& val)
{
    return val % 2 == 0;
}

int main()
{
    PRINT_HEADER("Start Program");

    bool (*select)(const int&) = nullptr;

    PRINT_HEADER("Create multidimensional array");

    int** pRagged = new int*[3];

    for (auto row = 0; row < 3; ++row)
    {
        pRagged[row] = new int[row + 10];
        for (auto col = 0; col < row + 10; ++col)
            pRagged[row][col] = row + col;
    }

    std::array<double> arr{};

    select = divBy2;
    copy_if(pRagged[2], pRagged[2] + 10, arr.begin(), select);

    PRINT_HEADER("Cleanup");

    delete[] pRagged;
    for (auto i = 2; i >= 0; --i)
        delete[] pRagged[i];
}
```

Use the editor to format your answer

Question 7

1 Point

The code below contains **two** (2) syntactic errors under C++17 standard (errors that are caught by a compiler or generate crashes/undefined behaviour at runtime). Your task is to identify each one by the file name and line number and **explain why the error appears, what C++ standard rule is broken, what C++ feature is misused and how the error should be fixed**. Your answer will be evaluated based on **clarity of the text** and the show of **understanding of the concepts** that are involved in the error.

```cpp
#include <iostream>
#include <array>
using namespace std;

#define PRINT_HEADER(msg) cout << "****  -> " << msg << " <-  ****\n"

bool divBy3(const int& val)
{
    return val % 3 == 0;
}

bool divBy2(const int& val)
{
    return val % 2 == 0;
}

int main()
{
    PRINT_HEADER("Start Program");

    bool (*select)(const double&) = divBy3;

    PRINT_HEADER("Create multidimensional array");

    int** pRagged = new int*[3];

    for (auto row = 0; row < 3; ++row)
    {
        pRagged[row] = new int[row + 10];
        for (auto col = 0; col < row + 10; ++col)
            pRagged[col][row] = row + col;
    }

    std::array<double, 10> arr{};

    select = divBy2;
    copy_if(pRagged[2], pRagged[2] + 10, arr.begin(), select);

    PRINT_HEADER("Cleanup");

    for (auto i = 2; i >= 0; --i)
        delete[] pRagged[i];
    delete[] pRagged;
}
```

Use the editor to format your answer

Question 8

1 Point

The code below contains **two** (2) syntactic errors under C++17 standard (errors that are caught by a compiler or generate crashes/undefined behaviour at runtime). Your task is to identify each one by the file name and line number and **explain why the error appears, what C++ standard rule is broken, what C++ feature is misused and how the error should be fixed**. Your answer will be evaluated based on **clarity of the text** and the show of **understanding of the concepts** that are involved in the error.

```cpp
#include <iostream>
#include <array>
using namespace std;

#define PRINT_HEADER(msg) cout << "****  -> " << msg << " <-  ****\n"

bool divBy3(const int& val)
{
    return val % 3 == 0;
}

bool divBy2(const int& val)
{
    return val % 2 == 0;
}

int main()
{
    PRINT_HEADER("Start Program");

    bool (*select)(const int&) = nullptr;

    PRINT_HEADER("Create multidimensional array");

    int* pRagged[3];

    for (auto row = 0; row < 3; ++row)
    {
        pRagged[row] = new int[row + 10];
        for (auto col = 0; col < row + 10; ++col)
            pRagged[row][col] = row + col;
    }

    std::array<double, 10> arr{};

    select = divBy2;
    copy_if(pRagged[2], pRagged[2] + 10, arr, select);

    PRINT_HEADER("Cleanup");

    for (auto i = 2; i >= 0; --i)
        delete[] pRagged[i];
    delete[] pRagged;
}
```

Use the editor to format your answer

Question 9

1 Point

The code below contains **two** (2) syntactic errors under C++17 standard (errors that are caught by a compiler or generate crashes/undefined behaviour at runtime). Your task is to identify each one by the file name and line number and **explain why the error appears, what C++ standard rule is broken, what C++ feature is misused and how the error should be fixed**. Your answer will be evaluated based on **clarity of the text** and the show of **understanding of the concepts** that are involved in the error.

```
 1. #include <iostream>
 2. #include <array>
 3. using namespace std;
 4.
 5. #define PRINT_HEADER(msg) cout << "****  -> "
 6.                                 << msg << " <-  ****\n"
 7.
 8. bool divBy3(const int& val)
 9. {
10.     return val % 3 == 0;
11. }
12.
13. bool divBy2(const int& val)
14. {
15.     return val % 2 == 0;
16. }
17.
18. int main()
19. {
20.     PRINT_HEADER("Start Program");
21.
22.     bool (*select)(const int&) = nullptr;
23.
24.     PRINT_HEADER("Create multidimensional array");
25.
26.     int** pRagged = new int*[3];
27.
28.     for (auto row = 0; row < 3; ++row)
29.     {
30.         for (auto col = 0; col < row + 10; ++col)
31.             pRagged[row][col] = row + col;
32.     }
33.
34.     std::array<double, 10> arr{};
35.
36.     select = divBy2;
37.     copy_if(pRagged[2], pRagged[2] + 10, arr.begin(), select);
38.
39.     PRINT_HEADER("Cleanup");
40.
41.     for (auto i = 2; i >= 0; --i)
42.         delete[] pRagged[i];
43.     delete[] pRagged;
44. }
```

Use the editor to format your answer

Question 10

1 Point

The code below contains **two** (2) syntactic errors under C++17 standard (errors that are caught by a compiler or generate crashes/undefined behaviour at runtime). Your task is to identify each one by the file name and line number and **explain why the error appears, what C++ standard rule is broken, what C++ feature is misused and how the error should be fixed**. Your answer will be evaluated based on **clarity of the text** and the show of **understanding of the concepts** that are involved in the error.

```cpp
#include <iostream>
#include <array>
using namespace std;

#define PRINT_HEADER(msg) cout << "****  -> " << msg << " <-  ****\n"

bool divBy3(const int& val)
{
    return val % 3 == 0;
}

bool divBy2(const int& val)
{
    return val % 2 == 0;
}

int main()
{
    PRINT_HEADER("Start Program");

    bool (*select)(const int&) = nullptr;

    PRINT_HEADER("Create multidimensional array");

    int** pRagged = new int[3];

    for (auto row = 0; row < 3; ++row)
    {
        pRagged[row] = new int[row + 10];
        for (auto col = 0; col < row + 10; ++col)
            pRagged[row][col] = row + col;
    }

    std::array<double, 10> arr{};

    select = divBy2;
    copy_if(pRagged[0].begin(), pRagged[0].end(), arr.begin(), divBy3);

    PRINT_HEADER("Cleanup");

    for (auto i = 2; i >= 0; --i)
        delete[] pRagged[i];
    delete[] pRagged;
}
```

Use the editor to format your answer

Questions Filter (10) ▼

Save and Close

Submit