YOU.I TV CODING CHALLENGE

PAGE ONE

```
#include <iostream>
#include <string>
#include <vector>
#include <memory>
class Initializable
{
public:
	Initializable()
	{
	Initialize();
	}
	~Initializable() {}
private:
	virtual void Initialize() {}
};
class Logger : public Initializable
{
public:
	Logger()
	: m_nLinesLogged(0)
	{
	}
	void Log(std::string logMessage) const
	{
	//MISTAKE A: Weird??? Sometimes the program doesn't log ANYTHING  (CLUE: Caused by the same problem as B)
	if (m_bInitialized)
	{
		std::cout << logMessage << std::endl;
		m_nLinesLogged++;
	}
	}
	int GetLinesLogged() const
	{
	return m_nLinesLogged;
	}
private:
	void Initialize() override
	{
	m_bInitialized = true;
	}
	bool m_bInitialized;
	mutable int m_nLinesLogged;
};
class Component : public Initializable
{
public:
	Component(int nComponentIndex)
	: m_pData(nullptr)
	{
	m_nComponentIndex = nComponentIndex;
	}
	struct ComponentData
	{
	std::string name;
	};
	int GetIndex() const
	{
	return m_nComponentIndex;
	}
	void Log(Logger *pLogger)
	{
	pLogger->Log("My index is: " + GetIndex());
	if (m_pData)
	{
		//MISTAKE B: We never see this log! (CLUE: Caused by the same problem as A)
		pLogger->Log("My name is: " + m_pData->name);
	}
	}
```

PAGE TWO

```
private:
	void Initialize() override
	{
	m_pData = new ComponentData();
	m_pData->name = std::string("Component #") + std::to_string(m_nComponentIndex + 1);
	}
	ComponentData *m_pData;
	int m_nComponentIndex;
};
class ComponentFactory
{
public:
	void SetLogger(Logger *pLogger)
	{
	m_pLogger.reset(pLogger);
	}

	std::vector<Component*> CreateComponents(int nCount)
	{
	std::vector<Component*> components;
	for (int i = 0; i < nCount; i++)
	{
		Component *pComponent = new Component(i);
		components.push_back(pComponent);
		pComponent->Log(m_pLogger.get());
	}
	return components;
	}
private:
	std::unique_ptr<Logger> m_pLogger;
};
int main()
{
	std::unique_ptr<Logger> pLogger(new Logger());
	{
	ComponentFactory factory;
	factory.SetLogger(pLogger.get());
	std::vector<Component*> components = factory.CreateComponents(10);
	//MISTAKE C: There's a crash if we uncomment this
	/*
		if (components[components.size()]->GetIndex() == 10)
		{
		pLogger->Log("Successfully created 10 components!");
		}
	*/
	}
	//MISTAKE D: We get a crash when the program exits
	//MISTAKE E: After we fixed the crashes, we ran a tool to detect memory leaks and found:
	//10 pointers leaking
	//10 integers leaking
	//MISTAKE F: The program output looks like this:
	/*
	My index is:
	y index is:
	index is:
	index is:
	ndex is:
	dex is:
	ex is:
	x is:
	is:
	is:
	*/
}
```