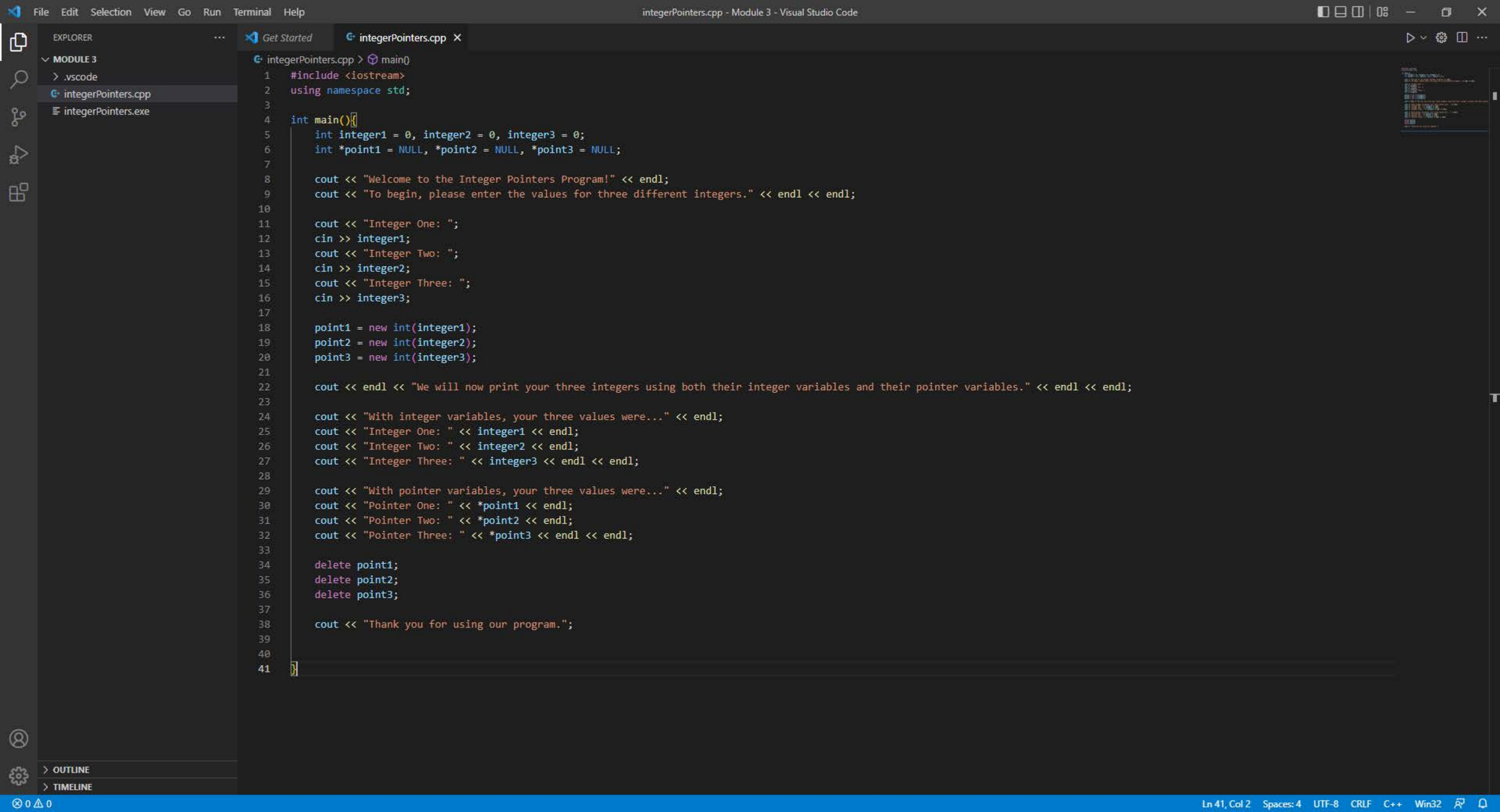
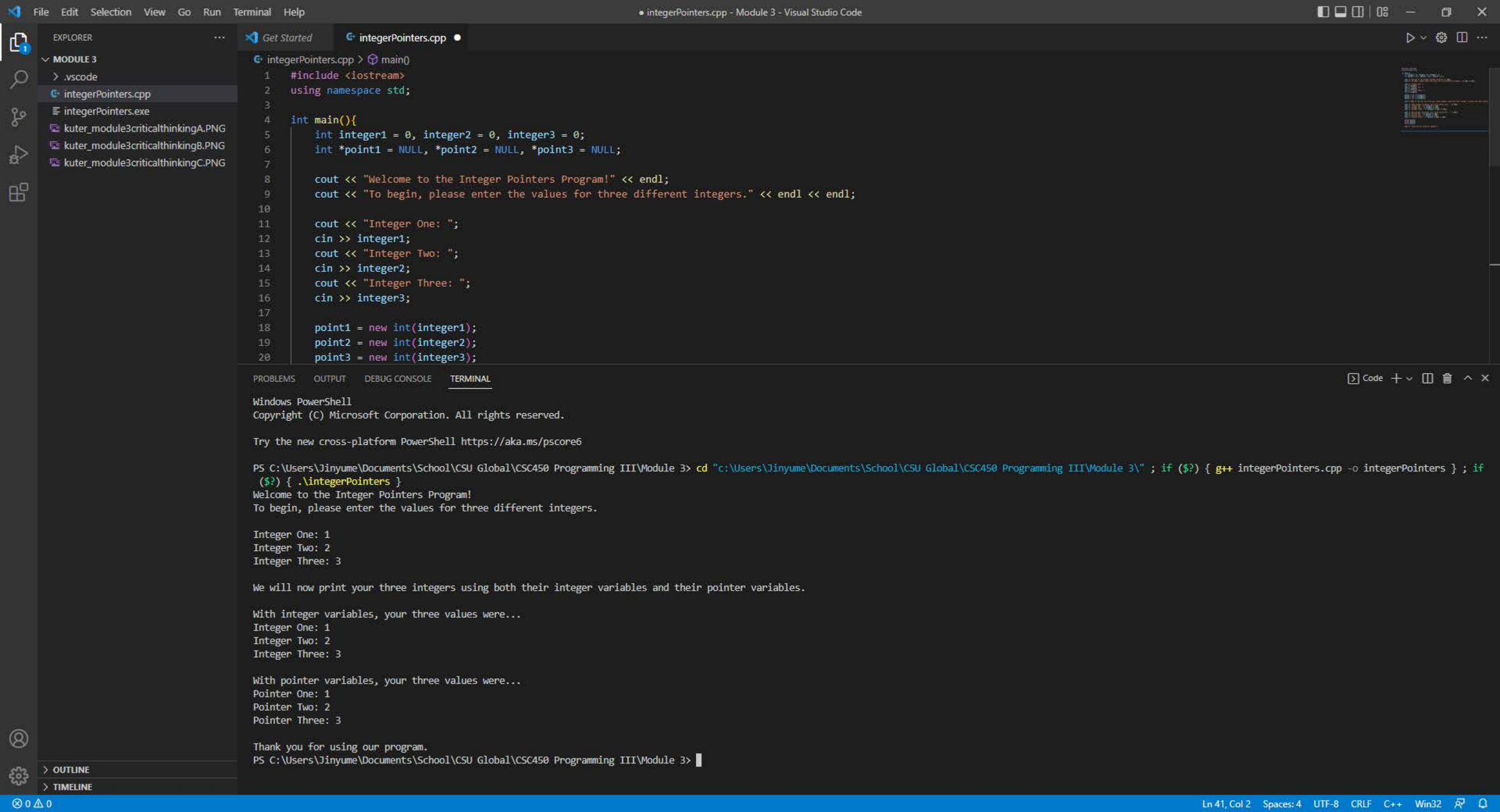

File Edit Selection View Go Run Terminal Help
integerPointers.cpp - Module 3 - Visual Studio Code
EXPLORER
MODULE 3
.vscode
integerPointers.cpp
integerPointers.exe
Get Started
integerPointers.cpp
integerPointers.cpp > main()
#include <iostream>
using namespace std;
int main(){
int integer1 = 0, integer2 = 0, integer3 = 0;
int *point1 = NULL, *point2 = NULL, *point3 = NULL;
cout << "Welcome to the Integer Pointers Program!" << endl;
cout << "To begin, please enter the values for three different integers." << endl << endl;
cout << "Integer One: ";
cin >> integer1;
cout << "Integer Two: ";
cin >> integer2;
cout << "Integer Three: ";
cin >> integer3;
point1 = new int(integer1);
point2 = new int(integer2);
point3 = new int(integer3);
cout << endl << "We will now print your three integers using both their integer variables and their pointer variables." << endl << endl;
cout << "With integer variables, your three values were..." << endl;
cout << "Integer One: " << integer1 << endl;
cout << "Integer Two: " << integer2 << endl;
cout << "Integer Three: " << integer3 << endl << endl;
cout << "With pointer variables, your three values were..." << endl;
cout << "Pointer One: " << *point1 << endl;
cout << "Pointer Two: " << *point2 << endl;
cout << "Pointer Three: " << *point3 << endl << endl;
delete point1;
delete point2;
delete point3;
cout << "Thank you for using our program.";
}
OUTLINE
TIMELINE
0 0
Ln 41, Col 2
Spaces: 4
UTF-8
CRLF
C++
Win32


File Edit Selection View Go Run Terminal Help
• integerPointers.cpp - Module 3 - Visual Studio Code
EXPLORER
MODULE 3
.vscode
integerPointers.cpp
integerPointers.exe
kuter_module3criticalthinkingA.PNG
kuter_module3criticalthinkingB.PNG
kuter_module3criticalthinkingC.PNG
Get Started
integerPointers.cpp
integerPointers.cpp > main()
#include <iostream>
using namespace std;
int main(){
int integer1 = 0, integer2 = 0, integer3 = 0;
int *point1 = NULL, *point2 = NULL, *point3 = NULL;
cout << "Welcome to the Integer Pointers Program!" << endl;
cout << "To begin, please enter the values for three different integers." << endl << endl;
cout << "Integer One: ";
cin >> integer1;
cout << "Integer Two: ";
cin >> integer2;
cout << "Integer Three: ";
cin >> integer3;
point1 = new int(integer1);
point2 = new int(integer2);
point3 = new int(integer3);
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
Code
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\Jinyume\Documents\School\CSU Global\CSC450 Programming III\Module 3> cd "c:\Users\Jinyume\Documents\School\CSU Global\CSC450 Programming III\Module 3\" ; if ($?) { g++ integerPointers.cpp -o integerPointers } ; if ($?) { .\integerPointers }
Welcome to the Integer Pointers Program!
To begin, please enter the values for three different integers.
Integer One: 1
Integer Two: 2
Integer Three: 3
We will now print your three integers using both their integer variables and their pointer variables.
With integer variables, your three values were...
Integer One: 1
Integer Two: 2
Integer Three: 3
With pointer variables, your three values were...
Pointer One: 1
Pointer Two: 2
Pointer Three: 3
Thank you for using our program.
PS C:\Users\Jinyume\Documents\School\CSU Global\CSC450 Programming III\Module 3>
OUTLINE
TIMELINE
Ln 41, Col 2 Spaces: 4 UTF-8 CRLF C++ Win32

EXPLORER

MODULE 3

- .vscode
- integerPointers.cpp
- integerPointers.exe
- kuter_module3criticalthinkingA.PNG
- kuter_module3criticalthinkingB.PNG

Get Started | integerPointers.cpp •

integerPointers.cpp > main()

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

int main(){
    int integer1 = 0, integer2 = 0, integer3 = 0;
    int *point1 = NULL, *point2 = NULL, *point3 = NULL;

    cout << "Welcome to the Integer Pointers Program!" << endl;
    cout << "To begin, please enter the values for three different integers." << endl << endl;

    cout << "Integer One: ";
    cin >> integer1;
    cout << "Integer Two: ";
    cin >> integer2;
    cout << "Integer Three: ";
    cin >> integer3;

    point1 = new int(integer1);
    point2 = new int(integer2);
    point3 = new int(integer3);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\Jinyume\Documents\School\CSU Global\CSC450 Programming III\Module 3> cd "c:\Users\Jinyume\Documents\School\CSU Global\CSC450 Programming III\Module 3\" ; if ($?) { g++ integerPointers.cpp -o integerPointers } ; if
 ($?) { .\integerPointers }
Welcome to the Integer Pointers Program!
To begin, please enter the values for three different integers.

Integer One: 5389
Integer Two: 123450
Integer Three: 43256224367

We will now print your three integers using both their integer variables and their pointer variables.

With integer variables, your three values were...
Integer One: 5389
Integer Two: 123450
Integer Three: 2147483647

With pointer variables, your three values were...
Pointer One: 5389
Pointer Two: 123450
Pointer Three: 2147483647

Thank you for using our program.
PS C:\Users\Jinyume\Documents\School\CSU Global\CSC450 Programming III\Module 3>
```

OUTLINE
TIMELINE

Clarissa Kuter

CSC450 Programming III

Reginald Haseltine

3 July 2022

Integer Pointers Program Analysis

For the critical thinking assignment due this week, I was asked to create another simple C++ console application. Like the previous assignment, this application was required to ask the user for three different inputs; however, unlike the previous assignment, this application required the user to input three integer values instead of three strings. While using new and delete operators to manage the memory, the program was required to store the user's input values into three different variables. I then needed to create an integer pointer to dynamic memory for each of the variables. Finally, the program was required to print the final values of the variables and pointer variables to the screen.

When toying with integer pointers, there are many things that could go awry. To avoid errors and vulnerabilities while playing with pointers, its important to start by initializing the pointer. If it is not initialized, the pointer will contain garbage. If the pointer contains garbage, it will not automatically be highlighted as a problem, but if it is called upon while containing garbage, that improper value can cause errors further along the application's run. Another important step to follow is to allocate memory for the points before assigning a value to it. If you assign a value to the pointer without allocating memory ahead of time, the program has the

potential of reading and writing the value to an invalid memory location. While the value being assigned to an unknown location may go unnoticed in smaller programs, it is still an issue that must be dealt with immediately. If left in the application, it can cause other important program data to be rewritten over, a segmentation fault to occur, and/or other mishaps to possibly occur.

Mishandling pointers may seem scary; however, that does not mean one should avoid using pointers in their code. They might have the potential to spark errors and create unusual bugs, but if the programmer double or triple checks where the pointer is pointing and what it is doing, they should work just fine. By learning how they work and the common misunderstandings that lead to their bugs, it will be easy to utilize pointers in one's program.