**Difference between plain enum and enum class**

//Difference between "enum" and "enum class"

#include "stdafx.h"

#include <iostream>

int main()

{

//plain enum//////////////////////////////////////////////

enum Color1 { red, green, blue };

Color1 c1 = red;

switch (c1) {

case red: std::cout << "red" << std::endl; break;

case green: std::cout << "green" << std::endl; break;

case blue: std::cout << "blue" << std::endl; break;

}

/\*3 problems with using plain enum

1. The red, green, blue variables take full scope. "int red=100;" would cause an error

2. Suppose you made another enum with the same variables. It also wouldn't compile, for reason above

3. The red, green, blue are saved as integers. However, this causes confusions

\*/

//enum class//////////////////////////////////////////////

enum class Color2 { red, green, blue };

Color2 c2 = Color2::green;

switch (c2) {

case Color2::red: std::cout << "red" << std::endl; break;

case Color2::green: std::cout << "green" << std::endl; break;

case Color2::blue: std::cout << "blue" << std::endl; break;

}

}

**Reason for using enum**

