

39 - Implicit Conversion and the Explicit Keyword in C++

quinta-feira, 6 de março de 2025 07:03

- Implicit means changing a word without telling the compiler to do so
 - o The compiler is allowed to do 1 implicit convert in your code
 - Without you having to cast it to convert from one data type to another
 - o Can simplify the code but better to avoid because it's confusing
 - o and looks a bit more easier
- Explicitly constructor, says that this constructor must be called to build this object
 - o Used when you want the constructor to be called as full, not allowing conversions
 - o If you don't want to convert numbers to vectors all the time

```
432 class Entity39_2
433 {
434 private:
435     std::string m_Name;
436     int m_Age;
437 public:
438     Entity39_2(const std::string& name): m_Name(name), m_Age(-1) {};
439     explicit Entity39_2(int age) : m_Name("Unknown"), m_Age(age) {};
440     // Entity39_2(const std::string& name, int age): m_Name(name), m_Age(age) {};
441     int getAge() { return m_Age; }
442 };
443
444 class Entity39
445 {
446 private:
447     std::string m_Name;
448     int m_Age;
449 public:
450     Entity39(const std::string& name): m_Name(name), m_Age(-1) {};
451     Entity39(int age): m_Name("Unknown"), m_Age(age) {};
452     Entity39(const std::string& name, int age): m_Name(name), m_Age(age) {};
453     int getAge() { return m_Age; }
454 };
455
456 void printEntity39(Entity39 e39){
457     std::cout << e39.getAge() << std::endl;
458 }
459
460 int main()
461 {
462     // explicitly constructors must be called to create an object, like Entity39_2.
463     // Must pass an valid variable type to the constructor. Otherwise it'll fail and will not try to implicitly convert it
464     // Just accept that type
465
466     Entity39_2 e39_1_2(22); // works fine because it's the explicit
467     Entity39_2 e39_2_2 = "Hugo"; // Fails because it's not called as explicit
468
469     // This is also allowed because it's a implicitly conversion, since we also have a constructor that supports is
470     // Entity39 e39_1("Hugo");
471
472     // This case we have omre than one conversion and the compiler an't work
473     // const char* -> Std::string -> Entity39
474     // Entity39 e39_1 = "Hugo";
475     Entity39 e39_2 = 28;
476
477     printEntity39(28);
478 }
```