

## Section 3.2 C++14

## auto Return

**Placeholder conversion functions**

The name of a conversion operator can be a placeholder. Multiple conversion operators can be defined in a single class, provided that no two have the same declared or deduced return type:

```
#include <cassert> // standard C assert macro

struct S
{
    static const int i;

    operator auto() { return 1; }
    operator long() { return 2L; }
    operator decltype(auto)() const { return (i); }
    operator const auto*() { return &i; }
};

const int S::i = 3;

void f1()
{
    S      s{};
    const S cs{};

    int      i1 = s; // Convert to int.
    long     i2 = s; // Convert to long.
    const int& i3 = s; // Convert to const int&.
    int      i4 = cs; // Convert to const int&.
    long     i5 = cs; // Convert to const int&.
    const int& i6 = cs; // Convert to const int&.
    long&     i7 = cs; // Error, cannot convert to long&
    const int* p1 = s; // Convert to int*.

    assert(1 == i1);
    assert(2L == i2);
    assert(3 == i3);
    assert(3 == i4);
    assert(3 == i5);
    assert(3 == i6);

    assert(p1 == &i3);
    assert(p1 == &i6);
}
```