7.17 Namespaces

Defining a namespace

A ***name conflict*** occurs when two or more items like variables, classes, or functions, have the same name. Ex: One programmer creates a Seat class for auditoriums, and a second programmer creates a Seat class for airplanes. A third programmer creating a reservation system for airline and concert tickets wants to use both Seat classes, but a compiler error occurs due to the name conflict.

A ***namespace*** defines a region (or scope) used to prevent name conflicts. Above, the auditorium seat class code can be put in an `auditorium` namespace, and airplane seat class code in an `airplane` namespace. The ***scope resolution operator ::*** allows specifying in which namespace to find a name, as in: `auditorium::Seat concertSeat;` and `airplane::Seat flightSeat;`.

PARTICIPATION ACTIVITY 7.17.1: Namespaces can resolve name conflicts.

1 2 3 2x speed

main.cpp

```
#include "auditorium.h"
#include "airplane.h"

int main() {
   auditorium::Seat concertSeat;
   airplane::Seat flightSeat;

   // ...

   return 0;
};
```

auditorium.h

```
namespace auditorium {
   class Seat {
      ...
   };
}
```

airplane.h

```
namespace airplane {
   class Seat {
      ...
   };
}
```

The two kinds of seats can then be declared as auditorium::Seat concertSeat and airplane::Seat flightSeat. The compiler now knows which Seat is which.

Feedback?

PARTICIPATION ACTIVITY 7.17.2: Namespaces.

1) Two same-named classes can cause a name conflict, but two same-named functions cannot.
   - ○ True
   - ◉ False

Correct

Same-named classes, functions, and even variables can cause a name conflict, even if the names are for different items. Ex: A class named Play and a function named Play could conflict.

2) A namespace helps avoid name conflicts among classes, functions, and other items in a program.
   - ◉ True
   - ○ False

Correct

By placing items in a namespace, a programmer can later specify which namespace to search for a named item. Especially in larger programs, and/or programs using code from many sources, the chances increase of identically named classes, functions, etc.

3) With namespaces, name conflicts cannot occur.
   - ○ True
   - ◉ False

Correct

Conflicts can still occur. Ex: If namespace airplane has both a Seat class and a Seat function, a name conflict may occur.

Feedback?

std namespace

All items in the C++ standard library are part of the ***std*** namespace (short for standard). To use classes like string or predefined objects like cout, a programmer can use one of two approaches:

1. ***Scope resolution operator (::):*** A programmer can use the scope resolution operator to specify the std namespace before C++ standard library items. Ex: `std::cout << "Hello";` or `std::string userName;`
2. ***Namespace directive***: A programmer can add the statement ***using*** `namespace std;` to direct the compiler to check the std namespace for any names later in the file that aren't otherwise declared. Ex: For `string userName;`, the compiler will check namespace std for string.

For code clarity, most programming guidelines discourage `using namespace` directives except perhaps for std.

PARTICIPATION ACTIVITY 7.17.3: std namespace.

1) Standard library items like

classes and functions are part of a namespace named std.

- (●) True
- ( ) False

Correct

The standard library already puts such items in a namespace named std. std is short for standard, meaning the standard library that is part of C++.

2) The namespace directive `using namespace std;` is required in any program.

- ( ) True
- (●) False

Correct

That line is just one option, and not required. Adding the line allows simpler code that accesses standard library items, for example allowing use of cout throughout code, rather than having to use the longer std::cout.

3) Without `using namespace std;`, a programmer can access cout using std::cout.

- (●) True
- ( ) False

Correct

The std:: part tells the compiler to look in namespace std for cout.

4) Without any namespace directive, `cout << num1` causes the compiler to check the std namespace for cout.

- ( ) True
- (●) False

Correct

The compiler does not automatically check the std namespace for names, and thus would report an undefined name error. If a programmer wants the compiler to check in the std namespace, the programmer should add the line: `using namespace std;`

Feedback?

CHALLENGE ACTIVITY 7.17.1: Enter the output from the proper namespace.

Jump to level 1

Type the program's output.

main.cpp | car.h | boat.h

```
#include "car.h"
#include "boat.h"
#include <iostream>

int main() {
   car::Dealership carDealer;
   boat::Dealership boatDealer;

   boatDealer.Sell();
   carDealer.Sell();
   boatDealer.Sell();

   cout << carDealer.GetStock() << " " << boatDealer.GetStock() << endl;

   return 0;

}
```

```
Sol
Sol
Sol
29
```

1

Check Next **Done**. Click any level to practice more. Completion is preserv

✔ Namespace boat defines a class named Dealership. Namespace car defines a different cla
`car::Dealership carDealer` declares a variable named carDealer of type Dealership from r

Yours

```
Sold boat
Sold car
Sold boat
29 18
```

Expected

```
Sold boat
Sold car
Sold boat
29 18
```

Feedback?