cout << "Two-day total: ";
show_time(trip);

travel_time day3 = {4, 32};
cout << "Three-day total: ";
show_time(sum(trip, day3));

return 0;
}

travel_time sum(travel_time t1, travel_time t2)
{
    travel_time total;

    total.mins = (t1.mins + t2.mins) % Mins_per_hr;
    total.hours = t1.hours + t2.hours +
                  (t1.mins + t2.mins) / Mins_per_hr;
    return total;
}
void show_time(travel_time t)
{
    cout << t.hours << " hours, "
        << t.mins << " minutes\n";
}

Here travel_time acts just like a standard type name; you can use it to declare variables, function return types, and function argument types. Because variables like total and t1 are travel_time structures, you can apply the dot membership operator to them. Note that because the sum() function returns a travel_time structure, you can use it as an argument for the show_time() function. Because C++ functions, by default, pass arguments by value, the show_time(sum(trip, day3)) function call first evaluates the sum(trip, day3) function call in order to find its return value. The show_time() call then passes sum()'s return value, not the function itself, to show_time(). Here's the program output:

Two-day total: 10 hours, 40 minutes
Three-day total: 15 hours, 12 minutes