C++ Utility Library - make_pair Function

Previous Page Next Page

Description

It constructs a pair object with its first element set to x and its second element set to y.

Declaration

Following is the declaration for std::make_pair function.

```
template <class T1, class T2>
   pair<T1,T2> make_pair (T1 x, T2 y);
```

C++11

```
template <class T1, class T2>
   pair<V1,V2> make_pair (T1&& x, T2&& y);
```

Parameters

x, y − These are two values.

Return Value

It returns a pair object whose elements first and second are set to x and y respectivelly.

Exceptions

Basic guarantee − if the construction or assignment of type T throws.

Data races

If either (or both) T1 or T2 is an rvalue reference type of a type supporting move semantics, its corresponding argument is modified.

Example

In below example explains about std::make_pair function.

```
#include <utility>
#include <iostream>

int main () {
   std::pair <int,char> foo;
   std::pair <int,int> bar;

   foo = std::make_pair (1,'A');
   bar = std::make_pair (100,3);

   std::cout << "foo: " << foo.first << ", " << foo.second << '\n';
   std::cout << "bar: " << bar.first << ", " << bar.second << '\n';

   return 0;
}
```

Let us compile and run the above program, this will produce the following result −

```
foo: 1, A
bar: 100, 3
```

Previous Page Next Page