

PairAndTuples

January 18, 2026

1 Pair and Tuples in C++

1.1 External Resources

- C++ Reference for Pair: <https://en.cppreference.com/w/cpp/utility/pair>
- C++ Reference for Tuple: <https://en.cppreference.com/w/cpp/utility/tuple>
- YouTube Video - <https://youtu.be/bA2ngITfSG8>
- YouTube Podcast - https://youtu.be/_KQVEhuGVDc
- NotebookLM learning materials - <https://notebooklm.google.com/notebook/d97a4380-9d69-4d19-a06f-a9648d0339c9>

1.2 Pair

- A **pair** in C++ is a simple container defined in the `<utility>` header that holds two values, which may be of different types. It is often used to group two related values together.
- a pair is a special case of a tuple that contains exactly two elements.
 - `first` and `second` members to access the elements.
- a pair is class template defined as `std::pair<T1, T2>`, where T1 and T2 are the types of the first and second elements, respectively.

```
#include <iostream>
#include <utility> // for std::pair
```

```
[1]: #include <iostream>
      #include <string>
      #include <utility> // for std::pair

      using namespace std;
```

```
[2]: // declare a pair
      pair<int, string> p1;
      cout << "first = " << p1.first << " second = " << p1.second << endl;
```

```
first = 0 second =
```

```
[3]: // can use get function too
      cout << " ( " << std::get<0>(p1) << ", " << std::get<1>(p1) << ")" << endl;
```

```
( 0, )
```

```
[4]: pair<int, string> p2(10, "Hello");
    cout << "first = " << p2.first << " second = " << p2.second << endl;
```

first = 10 second = Hello

```
[5]: pair<int, string> p2 = {10, "Hello"};
    cout << "first = " << p2.first << " second = " << p2.second << endl;
```

first = 10 second = Hello

```
[6]: pair<int, string> p2 = make_pair(10, "Hello");
    cout << "first = " << p2.first << " second = " << p2.second << endl;
```

first = 10 second = Hello

```
[7]: // tuple is mutable!
    p2.first = 20;
    p2.second = "World";
    cout << "first = " << p2.first << " second = " << p2.second << endl;
```

first = 20 second = World

```
[8]: // unpack a pair; use tie to bind values
    int num;
    string str;
    tie(num, str) = p2;
    cout << "num = " << num << " str = " << str << endl;
```

num = 20 str = World

```
[9]: // use structured bindings (C++17 and later) to unpack
    // doesn't work on Jupyter Notebooks currently
    // const auto [n, s] = p2;
    // cout << "num = " << n << " str = " << s << endl;
```

1.3 Tuple

- A **tuple** in C++ is a more general container defined in the `<tuple>` header that can hold a fixed number of elements, which may be of different types. It is used to group multiple related values together.
- A tuple can contain any number of elements, unlike a pair which is limited to two.
- A tuple is a class template defined as `std::tuple<Types...>`, where `Types...` represents a variadic list of types for the elements in the tuple.

```
[10]: #include <tuple> // for std::tuple
```

```
[11]: tuple<int, double, char, string> student1(1, 3.5, 'A', "Alice");
    tuple<int, double, char, string> student2 = {2, 3.2, 'C', "Charlie"};
    // use make_tuple non-member function
    tuple<int, double, char, string> student3 = make_tuple(3, 3.8, 'B', "Bob");
```

```
[12]: std::cout << "ID: " << std::get<0>(student1) << ", "
      << "GPA: " << std::get<1>(student1) << ", "
      << "grade: " << std::get<2>(student1) << ", "
      << "name: " << std::get<3>(student1) << '\n';
```

ID: 1, GPA: 3.5, grade: A, name: Alice

```
[13]: int id2;
      double gpa2;
      char grade2;
      string name2;
      tie(id2, gpa2, grade2, name2) = student2;
      std::cout << "ID: " << id2 << ", "
      << "GPA: " << gpa2 << ", "
      << "grade: " << grade2 << ", "
      << "name: " << name2 << '\n';
```

ID: 2, GPA: 3.2, grade: C, name: Charlie

1.4 Kattis Problems

- Testing LEDs - <https://open.kattis.com/problems/testingleds>
 - Hint: Use vector of pair to store the LED time and their state. sort and print the light turned off earliest
- Club Pizza - <https://open.kattis.com/problems/clubpizza>
 - Hint: Greedy algorithm. Use vector of pair of club time and pizza slices blaster can eat; sort the vector