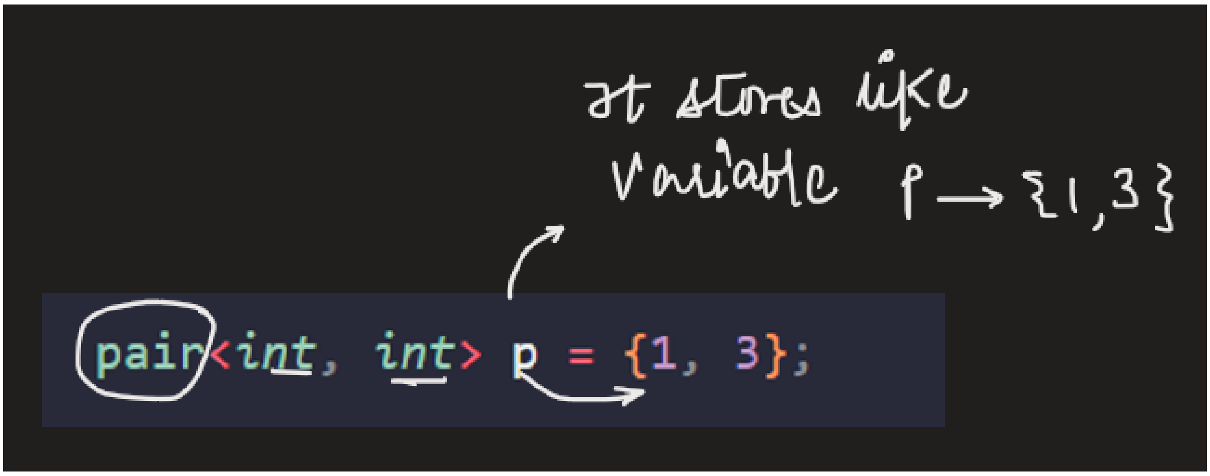
1️⃣

Pairs

Pairs

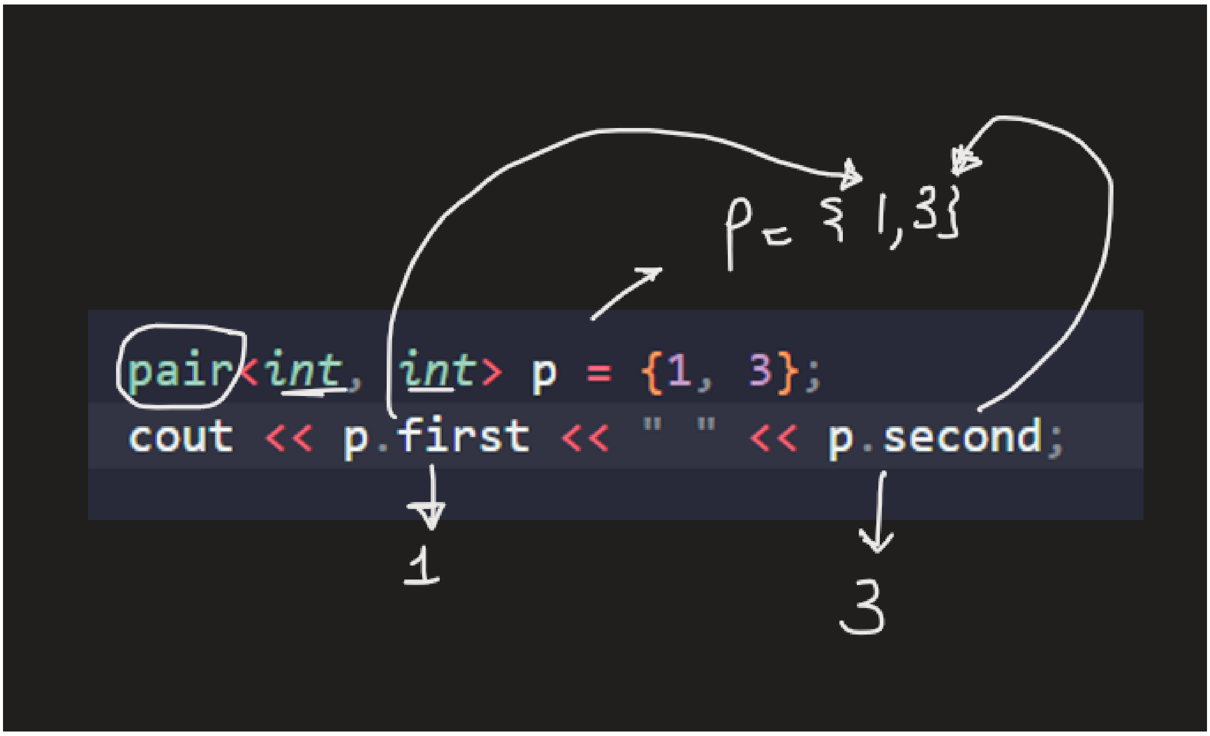
Pair is used to combine together two values that may be of different data types. Pair provides a way to store two heterogeneous objects as a single unit. It is basically used if we want to store tuples. The pair container is a simple container defined in **<utility>** header consisting of two data elements or objects.

- The first element is referenced as ‘first’ and the second element as ‘second’ and the order is fixed (first, second).
- Pair can be assigned, copied, and compared. The array of objects allocated in a **map** or hash_map is of type ‘pair’ by default in which all the ‘first’ elements are unique keys associated with their ‘second’ value objects.
- To access the elements, we use variable name followed by dot operator followed by the keyword first or second.



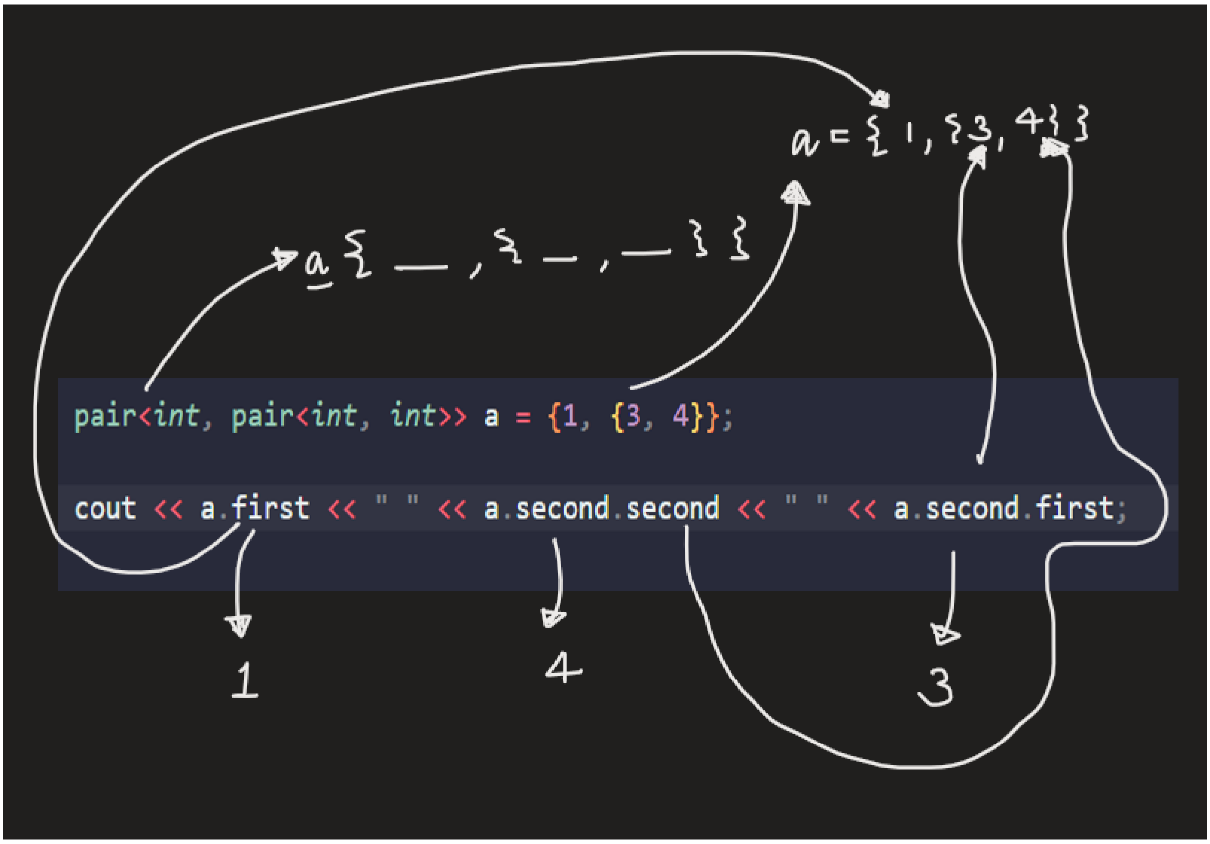
```cpp
pair<int, int> p = {1, 3};
```

How to declare the pair and initialize a value of pairs



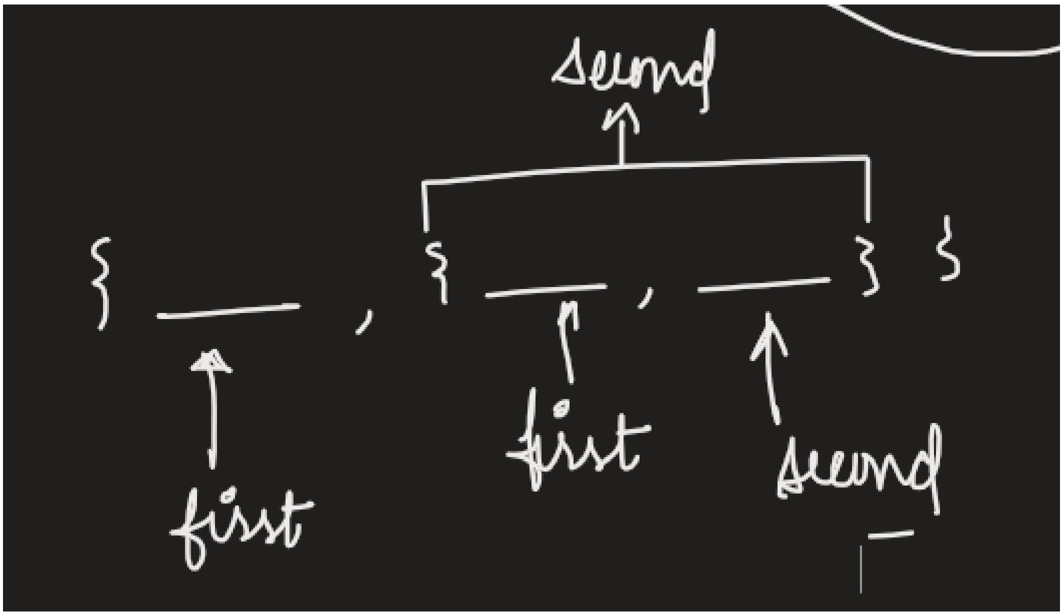
```cpp
pair<int, int> p = {1, 3};
cout << p.first << " " << p.second;
```

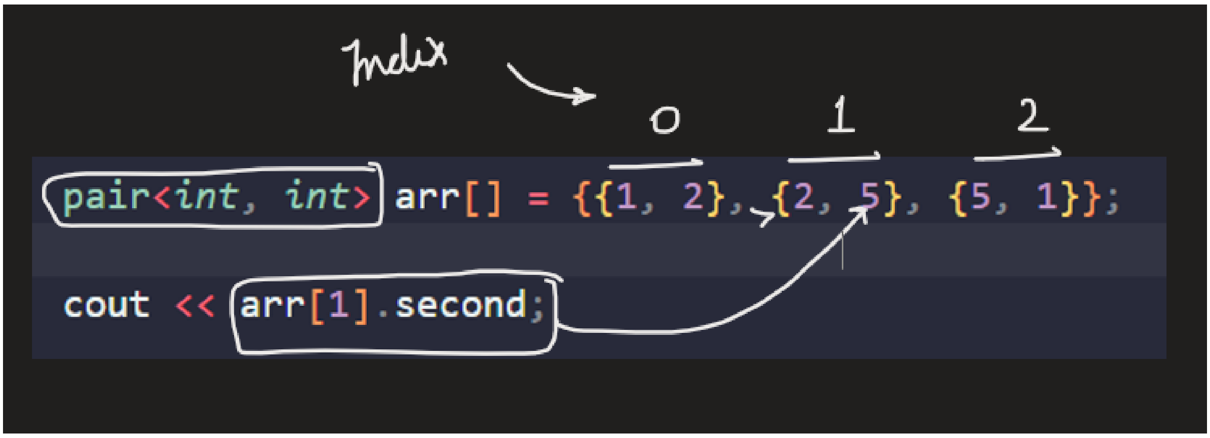
How to access the value of the pair



```cpp
pair<int, pair<int, int>> a = {1, {3, 4}};

cout << a.first << " " << a.second.second << " " << a.second.first;
```





```
pair<int, int> arr[] = {{1, 2}, {2, 5}, {5, 1}};

cout << arr[1].second;
```

How to declare and initialize array in a pair

The Whole Code for Pairs —

```
#include <iostream>
using namespace std;

// Pair is the part of the utility library
void explainPair()
{
      pair<int, int> p = {1, 3};
      cout << p.first << " " << p.second; // p.first --> 1 and p.second --> 3

      pair<int, pair<int, int>> a = {1, {3, 4}};

      cout << a.first << " " << a.second.second << " " << a.second.first;
      /*
      a.first --> 1
      a.second.second -->4
      a.second.first --> 3
      */

      pair<int, int> arr[] = {{1, 2}, {2, 5}, {5, 1}};

      cout << arr[1].second; // 5
}

int main()
{
      explainPair();
      return 0;
}
```