# Pair in C++

Hello coders,We welcome you all to learn a new concept on std::pair.

Generally, a pair is combination of two objects which had a well defined meaning .So let's jump in,

**Definition:**

The pair in the C++ standard template library is defined under utility library and it is used to store two data elements or objects of same or different datatype.

* To access first element of pair ,we can use the pair\_name followed by dot operator followed by a member variable "first" (member variable is declared inside the pair class ) and second element of the pair can be access using the pair\_name followed by dot operator followed by a member variable "second".
* Pair is heterogeneous i.e. it allows to store a variable of different data types.
* Pair can be assigned ,compared and copied.

Let's see some implementation of pair in the following code:

#include<iostream>

#include<utility>

using namespace std;

void print(pair<int,int> x)

{

cout<<"First element = "<<x.first

<<" "<<"second element = "<<x.second<<endl;

cout<<"--------------------------------------------------------------------"<<endl;

}

int main()

{

//declaring a pair of integers

pair<int, int > p;

//assign values to pair p

p.first=10;

p.second=100;

print(p);

//Another way of declaring and intializing a pair of two different datatype

//can you guess why we are not using print() to print pair p1

pair<string,int> p1("Algopoint",1);

cout<<"First element = "<<p1.first

<<" "<<"second element = "<<p1.second<<endl;

cout<<"--------------------------------------------------------------------"<<endl;

//initialize using another pair

pair<int,int > p3(p);

print(p3);

//Another way to initialize a pair is by using the make\_pair() function.

pair<int, int> p4=make\_pair(20,200);

print(p4);

//Another way to print the value of pair using get<index>(pair object)

//to access first element use index 0 and for second element use index 1

cout<<"First element = "<<get<0>(p1)

<<" "<<"second element = "<<get<1>(p1)<<endl;

cout<<"--------------------------------------------------------------------"<<endl;

}

2)Logical operators(=,==,!=.>=,<=) :

We can use these operators with the pairs

1)assignment operator(=): This operator assign the right pair object of into the left pair object .

pair& operator= (const pair& pr);

Member first of left pair object is assigned pr.first , and member second of left pair object is assigned pr.second.

2)equal operator(==): This operator compares the member variable of two pair objects.

Two pair objects compare equals and result a boolean true if pair1.first is equal to pair2.first and pair1.second is equal to pair2.second or else it returns false .

pair<int,int> p(10,100),p1(10,100);

if(p==p1)

{

cout<<"Both pairs are equal"<<endl;

}

else

cout<<"Both pairs are not equal"<<endl;

3)Not equal operator(!=): This operator return true if both of the member variable(i.e. p1.first != p2.first && p1.second!=p2.second) are not equal or return false.

pair<int,int> p(10,100),p1(10,100);

if(p!=p1)

{

cout<<"Both pairs are not equal"<<endl;

}

else

cout<<"Both pairs are equal"<<endl;

Similarly, operators <, >, <= and >= perform a [*lexicographical comparison*](http://www.cplusplus.com/lexicographical_compare)on the sequence formed by members first and second in all cases using operator<reflexively for the comparisons).

Let's see an example program on logical operator

#include<iostream>  
#include<utility>  
  
using namespace std;  
void print(pair<int,int> x)  
{  
 cout<<"First element = "<<x.first  
 <<" "<<"second element = "<<x.second<<endl;  
 cout<<"--------------------------------------------------------------------"<<endl;  
}  
  
int main()  
{  
 //declaring a pair p and another pair p1  
 pair<int,int> p(10,100),p1=p;  
 //p1=(10,100)  
 pair<int, int> p2(10,10);  
 print(p);  
 print(p1);  
 print(p2);  
 if(p==p1)cout<<"Both pairs are equal"<<endl;  
 if(p!=p2)cout<<"Both pairs are not equal"<<endl;  
 cout<<(p<=p1)<<endl;  
 cout<<(p>=p2)<<endl;  
 cout<<(p<p1)<<endl;  
 cout<<(p>p2)<<endl;  
 return 0;  
   
   
   
}

**Output:**

First element = 10 second element = 100  
--------------------------------------------------------------------  
First element = 10 second element = 100  
--------------------------------------------------------------------  
First element = 10 second element = 10  
--------------------------------------------------------------------  
Both pairs are equal  
Both pairs are not equal  
1  
1  
0  
1

3)Swap(): This function swaps the contents of one pair object with the contents of another pair object. The pairs must be of same type.

#include<iostream>  
#include<utility>  
  
using namespace std;  
void print(pair<int,int> x)  
{  
 cout<<"First element = "<<x.first  
 <<" "<<"second element = "<<x.second<<endl;  
 cout<<"--------------------------------------------------------------------"<<endl;  
}  
  
int main()  
{  
 //declaring a pair p and another pair p1  
 pair<int,int> p1(20,200);  
 //p1=(10,100)  
 pair<int, int> p2(10,100);  
 //printing the value of pair p1 before swapping  
 print(p1);  
 //swapping the content of pair p1 and pair p2  
 p1.swap(p2);  
 //printing the value of pair p1 after swapping  
 print(p1);  
 return 0;  
   
   
   
}

**Output:**

**First element = 20 second element = 200  
--------------------------------------------------------------------  
First element = 10 second element = 100  
--------------------------------------------------------------------**