for\_each loop in C++

Apart from the [generic looping techniques](https://www.geeksforgeeks.org/decision-making-c-c-else-nested-else/), such as “for, while and do-while”, C++ in its language also allows us to use another functionality which solves the same purpose termed “for-each” loops. This loop accepts a function which executes over each of the container elements. This loop is defined in the header file “algorithm”, and hence has to be included for successful operation of this loop.

**Why to use for\_each?**

* It is versatile, i.e can work with any container.
* It reduces chances of errors one can commit using generic for loop
* It makes code more readable
* for\_each loops improve overall performance of code

**Syntax:**

**for\_each (InputIterator first, InputIterator last, Function fn)**

**strt\_iter :** The **beginning** position

from where function operations has to be executed.

**last\_iter** : This **ending** position

till where function has to be executed.

**fnc/obj\_fnc :** The 3rd argument is a function or

an object function which operation would be applied to each element.

|  |
| --- |
| // C++ code to demonstrate the  // working of for\_each loop    #include<iostream>  #include<vector>  #include<algorithm>  using namespace std;    // helper function 1  void printx2(int a)  {      cout << a \* 2 << " ";  }    // helper function 2  // object type function  struct Class2  {      void operator() (int a)      {          cout << a \* 3 << " ";      }  } ob1;      int main()  {        // initializing array      int arr[5] = { 1, 5, 2, 4, 3 };        cout << "Using Arrays:" << endl;        // printing array using for\_each      // using function      cout << "Multiple of 2 of elements are : ";      for\_each(arr, arr + 5, printx2);        cout << endl;        // printing array using for\_each      // using object function      cout << "Multiple of 3 of elements are : ";      for\_each(arr, arr + 5, ob1);        cout << endl;        // initializing vector      vector<int> arr1 = { 4, 5, 8, 3, 1 };        cout << "Using Vectors:" << endl;          // printing array using for\_each      // using function      cout << "Multiple of 2 of elements are : ";      for\_each(arr1.begin(), arr1.end(), printx2);        cout << endl;        // printing array using for\_each      // using object function      cout << "Multiple of 3 of elements are : ";      for\_each(arr1.begin(), arr1.end(), ob1);        cout << endl;    } |

Run on IDE

Output:

Using Arrays:

Multiple of 2 of elements are : 2 10 4 8 6

Multiple of 3 of elements are : 3 15 6 12 9

Using Vectors:

Multiple of 2 of elements are : 8 10 16 6 2

Multiple of 3 of elements are : 12 15 24 9 3

**Exceptions and for\_each**

In the cases of exceptions, if the function throws an exception, **for\_each** loop will also throw an exception and **break/terminate** the loop.

|  |
| --- |
| // C++ code to demonstrate the working  // of for\_each with Exception    #include<iostream>  #include<vector>  #include<algorithm>  using namespace std;    // Helper function 1  void printx2(int a)  {      cout << a \* 2 << " ";      if ( a % 2 == 0)      {          throw a;      }    }    // Helper function 2  // object type function  struct Class2  {      void operator() (int a)      {          cout << a \* 3 << " ";          if ( a % 2 == 0)          {              throw a;            }      }  } ob1;      int main()  {        // Initializing array      int arr[5] = { 1, 5, 2, 4, 3 };        cout << "Using Array" << endl;        // Printing Exception using for\_each      // using function      try      {          for\_each(arr, arr + 5, printx2);      }      catch(int i)      {          cout << "\nThe Exception element is : " << i ;      }      cout << endl;        // Printing Exception using for\_each      // using object function      try      {          for\_each(arr, arr + 5, ob1);      }      catch(int i)      {          cout << "\nThe Exception element is : " << i ;      }        // Initializing vector      vector<int> arr1 = { 1, 3, 6, 5, 1 };        cout << "\nUsing Vector" << endl;        // Printing Exception using for\_each      // using function      try      {          for\_each(arr1.begin(), arr1.end(), printx2);      }      catch(int i)      {          cout << "\nThe Exception element is : " << i ;      }      cout << endl;        // printing Exception using for\_each      // using object function      try      {          for\_each(arr1.begin(), arr1.end(), ob1);      }      catch(int i)      {          cout << "\nThe Exception element is : " << i ;      }  } |

Run on IDE

Output:

Using Array

2 10 4

The Exception element is : 2

3 15 6

The Exception element is : 2

Using Vector

2 6 12

The Exception element is : 6

3 9 18

The Exception element is : 6