**Name – Shaurya Singh Srinet**

**Reg No – RA2111032010006**

**Branch –CSE w/s in IoT**

**Section – T2**

**Object Oriented Design and Programming**

**Assignment: Week 11: -**

1. Write a C++ Program to demonstrates Strings in STL.  
Code: -

#include <iostream>

#include <string>

using namespace std;

int main()

{

string str = "f335fd3b13fb2104f7fb2d429d9fb453", other\_str = "other string";

cout << "The string: " << str << "\n"

<< "The size allocated for the string: " << str.capacity() << "\n"

<< "The length of the string: " << str.length() << "\n";

str.resize(35);

cout << "After resize the size allocated for the string: " << str.capacity() << "\n";

str.shrink\_to\_fit();

cout << "Size of string after shrinking to fit: " << str.capacity() << "\n";

cout << "The reverse string using reverse iterator: ";

string::reverse\_iterator it;

for (it = str.rbegin(); it != str.rend(); it++)

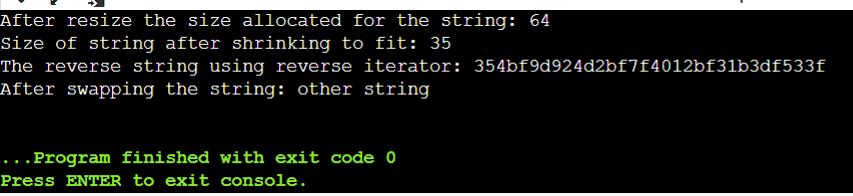
cout << \*it;

cout << "\n";

str.swap(other\_str);

cout << "After swapping the string: " << str << "\n";

return 0;}

OUTPUT-   


2. Write a C++ Program to demonstrates Arrays in STL.  
Code: -

#include <iostream>

#include <array>

using namespace std;

template <typename T, int n>

void print\_array(array<T, n> arr)

{

for (auto &element : arr)

{

cout << element << " ";

}

cout << "\n";

}

int main()

{

array<int, 5> arr = {1, 2, 3};

cout << "The element of the array: ";

print\_array<int, arr.size()>(arr);

arr.fill(115);

cout << "The elements of the array after filling: ";

print\_array<int, arr.size()>(arr);

arr[1] = 22;

arr[3] = 22;

cout << "The elements of the array after mutating: ";

print\_array<int, arr.size()>(arr);

cout << "The element arr[1]: " << arr.at(1) << "\n";

cout << "The element arr[2]: " << arr.at(2) << "\n";

return 0;

}

OUTPUT-   
Text

Description automatically generated

3. Write a C++ Program to demonstrates Lists in STL.  
Code: -

#include <iostream>

#include <list>

using namespace std;

template <typename T>

void print\_list(list<T> lst)

{

for (auto &num : lst)

{

cout << num << " ";

}

cout << "\n";

}

int main()

{

list<int> lst;

lst.push\_back(1);

cout << "The list after inserting a value at end: ";

print\_list(lst);

for (int i = 2; i <= 5; i++)

{

lst.push\_back(i \* i);

}

cout << "The list after inserting values: ";

print\_list(lst);

lst.push\_front(0);

cout << "The list after inserting a value at begin: ";

print\_list(lst);

lst.pop\_back();

cout << "The list after popping at end: ";

print\_list(lst);

lst.remove(4);

cout << "The list after removing a value: ";

print\_list(lst);

return 0;

}

OUTPUT-  
Text

Description automatically generated

4. Write a C++ Program to demonstrates Unordered\_set in STL  
Code: -

#include <iostream>

#include <unordered\_set>

using namespace std;

template <typename T>

void print\_set(unordered\_set<T> ust)

{

for (auto &element : ust)

{

cout << element << " ";

}

cout << "\n";

}

int main()

{

unordered\_set<int> ust = {1, 2, 1, 3, 5, 7};

cout << "The element of the unordered set: ";

print\_set(ust);

cout << "The size of the unordered set: " << ust.size() << "\n";

ust.clear();

cout << "The elements of the unordered set after clearing: ";

print\_set(ust);

ust.insert({2, 1, 5});

cout << "The elements of the unordered set after inserting values: ";

print\_set(ust);

ust.erase(2);

cout << "The elements of the unordered set after removing a value:";

print\_set(ust);

return 0;

}

OUTPUT-

Text

Description automatically generated

6. Write a C++ Program to demonstrates Pairs in STL.

Code: -

#include <iostream>

#include <utility>

using namespace std;

template <typename T, typename U>

void print\_pair(pair<T, U> p)

{

cout << p.first << ": " << p.second << "\n";

}

int main()

{

pair<string, int> math, chem, cs;

math.first = "Maths";

math.second = 95;

chem = make\_pair("Chemistry", 97);

cs = {"Computer Science", 100};

cout << "Marks\n";

print\_pair(math);

print\_pair(chem);

print\_pair(cs);

return 0;

}  
  
OUTPUT-

Text

Description automatically generated