**Practical 12**

**AIM**: Write a program to implement: Class template, Function template.

**OBJECTIVES:**

**Q.1) Implement a** **C++ program to display three different values and their size in bytes using function template.**

#include <iostream>

using namespace std;

template <class num>

num add(num a){

cout<<"x = "<<a<<endl;

cout<<"Size of "<<a<<" : "<<sizeof(a)<<endl;

}

int main()

{ int a=5;

add(a);

float b=10.3;

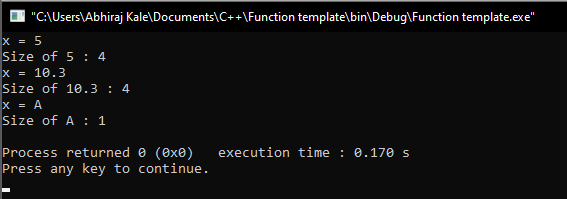
add(b);

char c='A';

add(c);

return 0;

}



**Q.2)** Implement a **C++ program to use class template. Create** a template class *sample* . It has two data a and b of generic types and two methods: *getdata()* to give input and *display()* to display data. Three object s1, s2 and s3 of this class is created. s1 operates on both integer data, s2 operates on one integer and another character data and s3 operates on one integer and another float data. Since, *sample*is a template class, it supports various data types.

#include <iostream>

using namespace std;

template <class T1 , class T2>

class Sample{

T1 a;

T2 b;

public:

void getdata(){

cout<<"Enter the 2 values :\n"; cin>>a;cin>>b;

}

void display(){

cout<<"Displaying values : "<<endl;

cout<<"a : "<<a<<endl;

cout<<"b : "<<b<<endl;

}

};

int main()

{ Sample <int,int>obj1;

cout<<"Two integer data"<<endl;

obj1.getdata();

obj1.display();

Sample <int,char>obj2;

cout<<"Integer and character data"<<endl;

obj2.getdata();

obj2.display();

Sample <int,float>obj3;

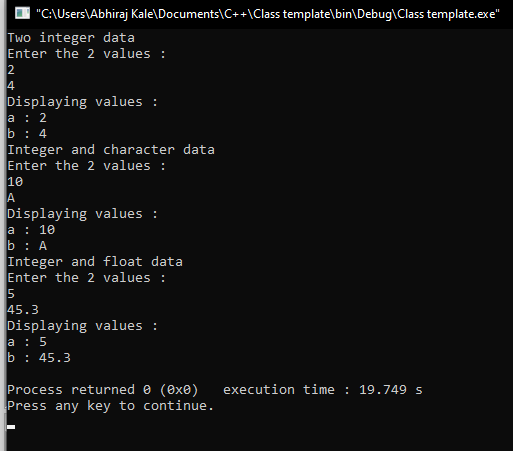
cout<<"Integer and float data"<<endl;

obj3.getdata();

obj3.display();

return 0;

}



**CONCLUSION:**

Hence, we have successfully written programs to implement class template and function template.