**Lab-6 Function Templates and Exception Handling**

**1. Write a program to show addition, subtraction, division, multiplication operations making four different functions. Test your program by using the concept of template handling.**

#include <iostream>

using namespace std;

template<class T1>

class Test

{

T1 a;

public:

void add (T1 x, T1 y)

{

a=x+y;

}

void mul(T1 x, T1 y)

{

a = x \* y;

}

void div(T1 x, T1 y)

{

a = x/y;

}

void sub(T1 x, T1 y)

{

a = x -y;

}

void show()

{

cout<<a<<"\n";

}

};

int main()

{

Test <float> testf;

Test <int> testi;

testf.add(5.23,6.43);testf.show();

testf.div(6.4,2.0);testf.show();

testi.mul(20,32); testi.show();

testi.sub(200,150); testi.show();

return 0;

}

**2. Write a program of class template with multiple parameters.**

#include<iostream>

using namespace std;

template <class T1, class T2>

class Test

{

T1 a;

T2 b;

public:

Test (T1 x, T2 y)

{

a = x;

b = y;

}

void show()

{

cout<<a<<" and "<<b<<"\n";

}

};

int main()

{

Test<float, int> test1(1.23,123);

Test<int,char>test2(100,'W');

test1.show();

test2.show();

return 0;

}

**3. Write a program to swap two numbers using function template.**

#include <iostream>

using namespace std;

template <class T>

void swap1(T &x, T &y)

{

T temp = x;

x = y;

y = temp;

}

void fun (int m, int n, float a, float b)

{

cout <<"m and n before swap: "<<m<<" "<<n<<"\n";

swap1 (m,n);

cout <<"m and n after swap: "<<m<<" "<<n<<"\n";

cout <<"a and b before swap: "<<a<<" "<<b<<"\n";

swap1(a,b);

cout <<"a and b after swap: "<<a<<" "<<b<<"\n";

}

int main()

{

fun(100,200,11.22,33.44);

return 0;

}

**4. Write a program to overload function template.**

#include<iostream>

#include<string.h>

using namespace std;

template<class T>

void display (T x)

{

cout<<”Template display:”<<x<<”\n”;

}

void display (int x)

{

cout<<”Explicity display:”<<x<<”\n”;

}

int main()

{

display (100);

display(12.34);

display (‘C’);

return 0;

}

**5. Write a program to show the basic concept of exception handling**

#include<iostream>

using namespace std;

int main()

{

int a,b;

cout<<"Enter values of a and b\n";

cin>>a>>b;

int x = a - b;

try{

if(x!=0)

{

cout<<"Result(a/x)="<<a/x<<"\n";

}

else{

throw(x);

}

}

catch(int i)

{

cout<<"Exception caught : x= "<<x<<"\n";

}

cout<<"END" ;

return 0;

}