Software Development Lab – II

Assignment Sheet

Week 9

Q1. Predict the output of following program.

template <typename T>

void abcd(const T&x)

{ static int count = 0;

cout << "x = " << x << " count = " << count << endl;

++count;

return;

}

int main()

{

abcd<int> (1);

cout << endl;

abcd<int>(1);

cout << endl;

abcd<double>(1.1);

cout << endl;

return 0;

}

x = 1 count = 0

x = 1 count = 1

x = 1.1 count = 0

Q2. Predict the output of the following program.

template <class T>

class Test

{ private:

T val;

public:

static int count;

Test() { count++; }

};

template<class T>

int Test<T>::count = 0;

int main()

{

Test<int> a;

Test<int> b;

Test<double> c;

cout << Test<int>::count << endl;

cout << Test<double>::count << endl;

return 0;

}

Output is

2

1

Q3. Create a template class calculator to perform addition, subtraction, multiplication and division of two numbers. Show the results for different datatypes.

#include<iostream>

using namespace std;

template <class t>

class calculator

{

private:

t num1,num2;

public:

calculator(t x,t y)

{

num1=x;

num2=y;

}

void display()

{

cout<< "Addition is: " << add() << endl;

cout << "Subtraction is: " << sub()<< endl;

cout << "Product is: " << mul() << endl;

cout << "Division is: " << divide() << endl;

}

t add(){return num1+num2;}

t sub(){return num1-num2;}

t mul(){return num1\*num2;}

t divide(){return num1/num2;}

};

int main()

{

int x,y;

cout<<"Enter the value of integer :";

cin>>x>>y;

calculator<int> obj1(2,3);

calculator<float> obj2(2.2,3.3);

cout<<"Int result is :"<<endl;

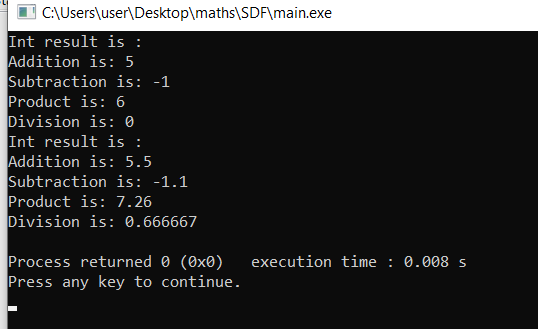
obj1.display();

cout<<"Int result is :"<<endl;

obj2.display();

return 0;

}



Q4. Write templates for the two functions, namely minimum and maximum. Minimum function should accept two arguments and return the value of the arguments that is the lesser among the two. Maximum function should accept two arguments and return the value of the arguments that is the greater among the two values. Design a simple driver program that demonstrates the templates with various data types.

#include<iostream>

using namespace std;

template<class x>

x Minimum(x a,x b){

if(a<b)

return a;

return b;

};

template<class x>

x Maximum(x a, x b){

if(a>b)

return a;

return b;

}

int main(){

int var,var1;

cout<<"Enter two values of int type: ";

cin>>var>>var1;

cout<<"Minimum value is:"<<Minimum(var,var1);

cout<<endl<<"Maxumum value is:"<<Maximum(var,var1);

cout<<"\nEnter two values of float type: ";

float var2,var3;

cin>>var2>>var3;

cout<<"Minimum value is:"<<Minimum(var2,var3);

cout<<endl<<"Maxumum value is:"<<Maximum(var,var1);

cout<<"\nEnter two values of double type: ";

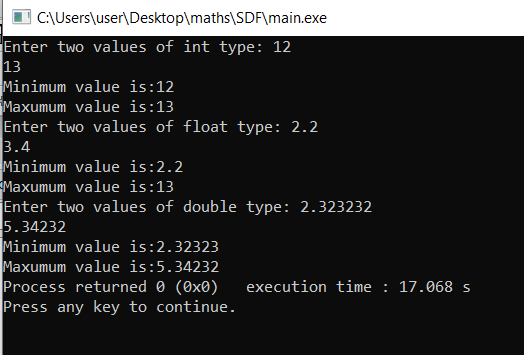
double var4,var5;

cin>>var4>>var5;

cout<<"Minimum value is:"<<Minimum(var4,var5);

cout<<endl<<"Maxumum value is:"<<Maximum(var4,var5);

}



Q5. Create your own template class MyVector with data members and member functions such that the size(), push\_back() and pop\_back() functionalities of Vector can also be performed by MyVector.

Q6. Implement the following problem using vector STL in c++. Consider two arrays of similar type, having different numbers of elements. Take the array values from user till a negative value for both the arrays. Insert those elements of second array into first array which are not present in the first array. Also display the first array after insertion in sorted form.