**Session 10 (unit-6): Generic Programming**

1. **WAP to display the largest among two numbers using function templates.**

#include <iostream>

using namespace std;

template <class T>

T Largest(T a, T b)

{

return ( a > b) ? a : b;

}

int main()

{

int x1, x2;

float y1, y2;

char z1, z2;

cout << "Enter two integers: "<<endl;

cin >> x1 >> x2;

cout << Largest(x1, x2) <<" is larger." << endl;

cout << "\n Enter two floating-point numbers: \n";

cin >> y1 >> y2;

cout << Largest(y1, y2) <<" is larger." << endl;

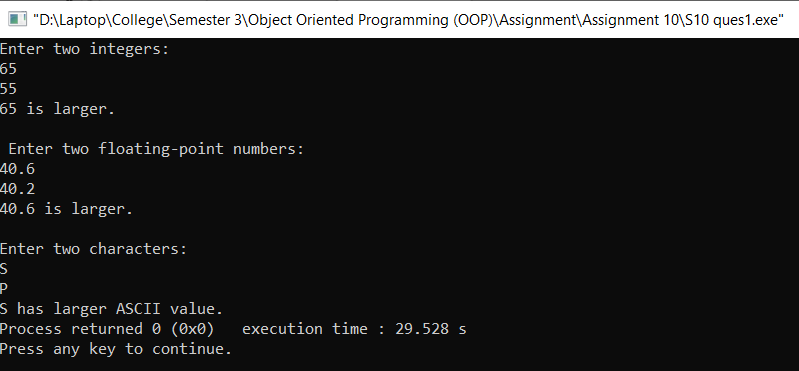
cout << "\nEnter two characters:\n";

cin >> z1 >> z2;

cout << Largest(z1, z2) << " has larger ASCII value.";

return 0;

}



1. **WAP to swap data using function templates.**

#include<iostream>

using namespace std;

template< class T>

T Swap(T &a, T &b)

{

T temp;

temp = a;

a = b;

b = temp;

}

int main()

{

int x, y;

cout<<" Enter First Number ";

cin>>x;

cout<<"\n Enter Second Number ";

cin>>y;

cout << "Before passing data to function template.\n";

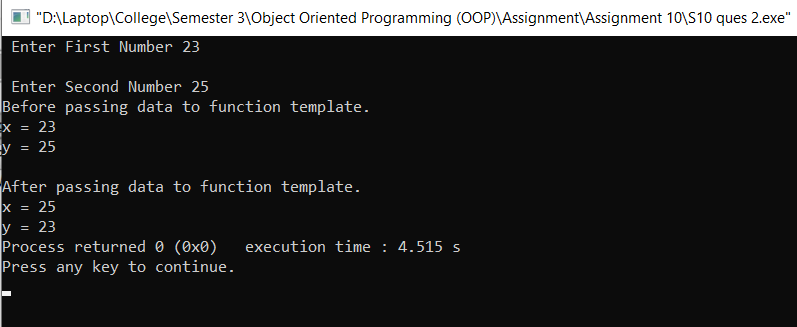
cout << "x = " << x << "\ny = " << y;

Swap(x, y);

cout << "\n\nAfter passing data to function template.\n";

cout << "x = " << x << "\ny = " << y;

}



1. **WAP that accepts two different data-types as arguments to the template function and returns the value.**

#include <iostream>

using namespace std;

template <class T, class X>

T value(T x, X y)

{

cout<<"\nThe variables are: "<<x<<" "<<y;

return 0;

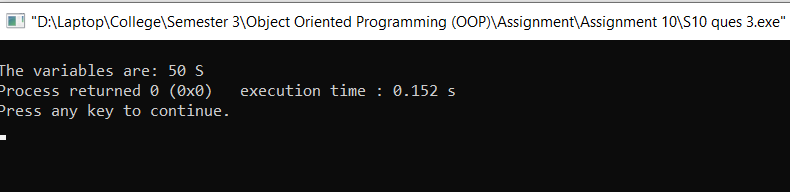
}

int main()

{

value(50,'S');

return 0;

}  
  


1. **WAP to add, subtract, multiply and divide two numbers using class template.**

#include <iostream>

using namespace std;

template<class T>

class A

{

private:

T a,b;

public :

T add(T a, T b)

{

return a + b;

}

T sub(T a, T b)

{

return a - b;

}

T mul(T a, T b)

{

return a \* b;

}

T div(T a, T b)

{

return a / b;

}

};

int main()

{

A<int> ob;

int x,y;

cout<<"Enter the values ";

cin>>x>>y;

cout<<"Addition of "<<x<<" and "<<y<<" is: "<<ob.add(x,y)<<endl;

cout<<"Subtraction of "<<x<<" and "<<y<<" is: "<<ob.sub(x,y)<<endl;

cout<<"Multiplication of "<<x<<" and "<<y<<" is: "<<ob.mul(x,y)<<endl;

cout<<"Division of "<<x<<" and "<<y<<" is: "<<ob.div(x,y)<<endl;

return 0;

}

