CPP code

#include <iostream>

using namespace std;  
/\*  
int findMin(int a, int b){  
if(a<b){  
return a;  
}  
else{  
return b;  
}  
}

char findMin(char a, char b){  
if(a<b){  
return a;  
}  
else{  
return b;  
}  
}  
double findMin(double a, double b){  
if(a<b){  
return a;  
}  
else{  
return b;  
}  
}  
\*/  
//Using templates - help in writing function for all data type  
//during compile time T is replaced by the data type specified

template <typename T>  
T findMin(T a, T b){  
if(a<b){  
return a;  
}  
else{  
return b;  
}  
}

template <typename T1, typename T2>  
T1 getDetails(T1 a, T2 b){  
cout << "Datatype one is " << a << endl;  
cout << "Other one is " << b << endl;  
return a;  
}

//Generic class  
template <class T3> //can use class also  
class Example{  
T3 a,b;  
public:  
Example(T3 x, T3 y){  
a=x;  
b=y;  
}  
T3 findMin(){  
return (a<b)?a:b;  
}  
};

int main()  
{  
cout << findMin<int>(10,20) << endl;  
cout << findMin<double>(10.5,20.5) << endl;  
cout << findMin<char>('a','s') << endl;  
cout << getDetails<int, char>(12, 'c') << endl;  
Example<int> obj(10,20);  
cout << obj.findMin() << endl;  
Example<double> obj1(10.0,-20.5);  
cout << obj1.findMin();  
return 0;  
}

output:

