

ΚΛΗΡΟΝΟΜΙΚΟΤΗΤΑ – ΥΠΕΡΦΟΡΤΩΣΗ ΜΕΘΟΔΩΝ

Superclass Instrument.hpp

```
#include<iostream>
using namespace std;

class Instrument{ //Δήλωση υπερκλάσης
protected:
    double value;
public:

    Instrument(){ //Constructor χωρίς όρισμα
        value = 0;
    }
    Instrument(double v){ //Constructor με όρισμα
        value = v;
    }
    ~Instrument(){ //Destructor

    }

    void insertValue(double aValue){ //Μέθοδος απόδοσης τιμής μέσω παραμέτρου
        value = aValue;
    }

    double returnValue(){ //Μέθοδος επιστροφής τιμής
        return value;
    }

    void printValue(){ //Μέθοδος εμφάνισης τιμής
        cout << "===Instrument Class Object===" << endl;
        cout << "The Value is: " << value << endl << endl;
    }
}; //Τέλος υπερκλάσης
```

Subclass HotTemperatures.hpp

```
#include<iostream>
#include "Instrument.hpp" //Συμπεριλαμβάνεται η υπερκλάση Instruments

class HotTemperatures:public Instrument{ //Δήλωση υποκλάσης
public:
    //Μέθοδος απόδοσης τιμής μέσω παραμέτρου με υπερφόρτωση
    void insertValue(double hotValue){
        if (hotValue > 0){
            Instrument::insertValue(hotValue); //Κλήση μεθόδου υπερκλάσης
            cout << hotValue << "oC is an Acceptable temperature " << endl;
        }
        else
            cout << hotValue << "oC Temperature is not acceptable because it is below zero!!"
<< endl;
    }

    //Μέθοδος εμφάνισης τιμής με υπερφόρτωση
    void printValue(){
        cout << "\n===Hot Value Class Object===" << endl;
        cout << "The Temperature is: " << value << endl << endl;
    }
}; //Τέλος υποκλάσης
```

Subclass ColdTemperatures.hpp

```
#include<iostream>
#include "Instrument.hpp" //Συμπεριλαμβάνεται η υπερκλάση Instruments

class ColdTemperatures:public Instrument{ //Δήλωση υποκλάσης
public:
    //Μέθοδος απόδοσης τιμής μέσω παραμέτρου με υπερφόρτωση
    void insertValue(double coldValue){
        if (coldValue < 0){
            Instrument::insertValue(coldValue); //Κλήση μεθόδου υπερκλάσης
            cout << coldValue << "oC is an Acceptable temperature " << endl;
        }
        else
            cout << coldValue << "oC Temperature is not acceptable because it is above zero!!"
<< endl;
    }

    //Μέθοδος εμφάνισης τιμής με υπερφόρτωση
    void printValue(){
        cout << "\n===Cold Value Class Object===" << endl;
        cout << "The Temperature is: " << value << endl << endl;
    }
}; //Τέλος υποκλάσης
```

Main Program

```
#include <iostream>
#include "HotTemperatures.hpp" //Συμπεριλαμβάνεται η υποκλάση HotTemperatures
#include "ColdTemperatures.hpp" //Συμπεριλαμβάνεται η υποκλάση ColdTemperatures
using namespace std;

int main(){
    cout << " =====INSTRUMENT OBJECT===== " << endl;
    Instrument Instr; //Αντικείμενο υπερκλάσης
    Instr.insertValue(100); //Απόδοση τιμής
    Instr.printValue(); //Εμφάνιση τιμής
    cout << endl;

    cout << " =====HOT TEMPERATURE OBJECT===== " << endl;
    HotTemperatures HotTemp; //Αντικείμενο υποκλάσης

    //Προσπάθεια απόδοσης μη αποδεκτής τιμής σε υπερφορτωμένη μέθοδο
    HotTemp.insertValue(-10);

    //Εμφάνιση τιμής με υπερφορτωμένη μέθοδο
    HotTemp.printValue();

    //Απόδοση αποδεκτής τιμής σε υπερφορτωμένη μέθοδο
    HotTemp.insertValue(200);

    //Εμφάνιση τιμής με υπερφορτωμένη μέθοδο
    HotTemp.printValue();
    cout << endl;

    cout << " =====COLD TEMPERATURE OBJECT===== " << endl;
    ColdTemperatures ColdTemp; //Αντικείμενο υποκλάσης

    //Προσπάθεια απόδοσης μη αποδεκτής τιμής σε υπερφορτωμένη μέθοδο
    ColdTemp.insertValue(20);

    //Εμφάνιση τιμής με υπερφορτωμένη μέθοδο
    ColdTemp.printValue();
```

```
//Απόδοση αποδεκτής τιμής σε υπερφορτωμένη μέθοδο
ColdTemp.insertValue(-20);

//Εμφάνιση τιμής με υπερφορτωμένη μέθοδο
ColdTemp.printValue();

cout << " =====END OF PROGRAM===== " << endl;
return 0;
} //Τέλος προγράμματος
```