Name : **Nithish.G**

REG no : **19BCS0012**

Course : **OBJECT ORIENTED PRORAMMING**

Course code : **CSC2002**

Assignment : **DIGITAL ASSIGNMENT-3**

2. Develop an OOP to check the whether the son and father is Senior Citizen by enabling the "checkSeniorCitizen()" function in both "Father" class and "Son" class using virtual functions concept.

Source code

```
#include<iostream>

using namespace std;

#include<conio.h>

class father

{

    public :

        int f;

        void fatherage()

        {

            cout<<"enter the father age  : " << endl;

            cin>>f;

        }

        virtual void c()

        {

             if(f>=60)

            {

                    cout<<" THE FATHER AGE IS : "<<f<<" SO HE IS SENIORCITIZEN "<<endl;

            }

            else

            {

               cout<<" THE FATHER AGE IS : "<<f<<"  SO HE IS NOT SENIORCITIZEN "<<endl;

            }

        }
```

```
};
class son:public father
{
    public:
        int s;
         void sonage()
        {
            cout<<"enter the son age  : " << endl;


            cin>>s;
         }
        void c()
        {
             if(s>=60)
            {
                    cout<<" THE SON AGE IS : "<<s<<"  SO HE IS SENIORCITIZEN "<<endl;
            }
            else
            {
               cout<<" THE SON AGE IS : "<<s<<" SO HE IS NOT SENIORCITIZEN "<<endl;
            }
        }
};


int main()
{
   father ob1,*p1;
```

```
    son ob2;
    ob1.fatherage();
    ob2.sonage();
    p1=&ob1;
    p1-> c();
    p1 =&ob2;
    p1-> c();
getch();
return 0;

}
```

3. Develop an OOP to perform the selection sort in ascending and descending by using overloaded function template.

```
// CPP program to sort array of any data types.
#include <bits/stdc++.h>
using namespace std;

// Template formed so that sorting of any
// type variable is possible
template<class T>
voidsortArray(T a[], int n)
{
        // boolean variable to check that
        // whether it is sorted or not
        bool b = true;
        while (b) {
                b = false;
```

```
                for (size_t i=0; i<n-1; i++) {


                        // swapping the variable
                        // for sorting order
                        if (a[i] > a[i + 1]) {
                                T temp = a[i];
                                a[i] = a[i + 1];
                                a[i + 1] = temp;
                                b = true;
                        }
                }
        }
}

// Template formed so that sorting of any
// type variable is possible
template<class T>
voidprintArray(T a[], int n)
{
        for (size_t i = 0; i < n; ++i)
                cout<< a[i] << " ";
        cout<<endl;
}

// Driver code
int main()
{
        int n = 4;
```

```
        intintArr[n] = { 2000, 456, -10, 0 };
        sortArray(intArr, n);
        printArray(intArr, n);

        stringstrArr[n] = { "We do nothing",
                                        "Hi I have something",
                                        "Hello Join something!",
                                        "(Why to do work)" };
        sortArray(strArr, n);
        printArray(strArr, n);

        floatfloatArr[n] = { 23.4, 11.4, -9.7, 11.17 };
        sortArray(floatArr, n);
        printArray(floatArr, n);

        return 0;
}
```

Output:

```
-10   0   456   2000
-9.7   11.17   11.4   23.4
```