// Programming Techniques

Exercises Class # 5

Master Programme in Mathematical Finance

1st Semester 2018/2019

ISEG-UL

Sara Lopes

{

//sblopes@iseg.ulisboa.pt

//saradutralopes@gmail.com

}

Overview

- Scope of Variables
- Call by Value vs Call by Reference
- Structures

Bibliography:

- Stroustrup, Bjarne, Programming Principles and Practice Using C++, Second Edition, Addison-Wesley, 2014.

Local and Global Variables

local: Variables defined in a block of code. Are defined only in that block of sub-blocks.

global: Variables defined outside functions.

```
double y=1;
double x;
cin>>x;
if(x>=0){
double z=sqrt(x);
y=x+z;
}
cout<<y<<endl; //valid
cout<<z<<endl; //invalid
```

Qualifiers

extern: Global variables are valid in every part of the code and can be accessed in different files. They can only be defined in one file and in the other files they need to be qualified as extern.

static: When this qualifier is applied to a global variable means that they can only be used in that file. When the qualifier is applied to local variables it means that the value of the variable is only created once and that the variable is not destroyed after execution.

const: The value of a variable qualified as const can not be changed.

Call by Value vs Call by Reference

- call by value : double **f(double x)** - creates a copy of x when f is called
- call by reference: **double f(double & x)** pass a reference as argument
- call by const reference: **double f(const double & x)** pass a reference and doesn't change the value

What happens to the value of x?

```
void f(int x){
x=4;
}
int main(){
int x=0;
f(x);
cout<<x;
return 0;
}
```

What happens to the value of x?

```
void f(int &x){
x=4;
}
int main(){
int x=0;
f(x);
cout<<x;
return 0;
}
```

What happens to the value of x?

```
void f(const int &x){
x=4;
}
int main(){
int x=0;
f(x);
cout<<x;
return 0;
}
```

Structure Example

Structure is a collection of variables of different types under a single name.

```
struct Student{

    char name[50];
    double gradet1;
    double gradet2;

};
void displayStudent(Student s){
    cout<<"Student's␣name:␣"<<s.name<<endl;
    cout<<"First␣project␣grade:␣"<<s.gradet1<<endl;
    cout<<"Second␣project␣grade:"<<s.gradet2<<endl;
}

double computefinalgrade(Student s){
    double grade=s.gradet1*0.4+s.gradet2*0.6;
    return grade;
}
```

Structure Example

```
int main()
{
Student s1;

strcpy(s1.name,"John");
s1.gradet1=17;
s1.gradet2=18;

displayStudent(s1);
cout<<s1.name<<"␣final␣grade␣is:␣"<<computefinalgrade(s1);

 return 0;
}
```