National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Computing

CS212: Object Oriented Programming

Class: BSCS-5AB

Lab 4: Classes & Static/Const Data Members

Date: Mar 7, 2016

Time: 9am – 12pm / 2pm to 5pm

Instructor: Ms. Hirra Anwar

Lab 4: Classes & Static/Const Data Members

Introduction

In this lab you will learn about passing objects as arguments & Static/Const Data Members.

Objectives

To understand the use of Classes & Static/Const Data Members.

Tools/Software Requirement

Visual Studio 2010/2012

Description

The first example is related to passing objects as arguments and the creation of class/constructors/member functions, to revise the previous concepts.

Pass object as an arguments to class

Lab Practice **1:**

```
/*C++ PROGRAM TO PASS OBJECT AS AN ARGUMEMT. The program Adds the two heights given in feet and inches. */

#include< iostream.h>
#include< conio.h>

class height
{
int feet,inches;
public:
void getht(int f,int i)
{
feet=f;
inches=i;
}
void putheight()
{
cout< < "\nHeight is:"< < feet< < "feet\t"< < inches< < "inches"< < endl;
}
void sum(height a,height b)
```

```
{
height n;
n.feet = a.feet + b.feet;
n.inches = a.inches + b.inches;
if(n.inches ==12)
{
n.feet++;
n.inches = n.inches -12;
}
cout< < endl< < "Height is "< < n.feet< < " feet and "< < n.inches< < endl;
}
};
void main()
{
height h,d,a;
clrscr();
h.getht(6,5);
a.getht(2,7);
h.putheight();
a.putheight();
d.sum(h,a);
getch();
}
```

```
/**********OUTPUT***********

Height is:6feet 5inches

Height is:2feet 7inches

Height is 9 feet and 0
```

Static Functions/Members

By declaring a function member as static, you make it independent of any particular object of the class. A static member function can be called even if no objects of the class exist and the **static** functions are accessed using only the class name and the scope resolution operator **::**.

A static member function can only access static data member, other static member functions and any other functions from outside the class.

Static member functions have a class scope and they do not have access to the **this** pointer of the class. You could use a static member function to determine whether some objects of the class have been created or not.

Static keyword should only be used in the declaration and not in the definition of data members or functions.

Let us try the following example to understand the concept of static function members:

Lab Practice Example

```
#include <iostream>

using namespace std;

class Box
{
  public:
    static int objectCount;
    // Constructor definition
    Box(double l=2.0, double b=2.0, double h=2.0)
    {
      cout <<"Constructor called." << endl;
      length = l;
      breadth = b;
      height = h;
      // Increase every time object is created
      objectCount++;
    }
    double Volume()
    {
      return length * breadth * height;
    }
    static int getCount()
    {
      return objectCount;
    }
  private:
    double length;     // Length of a box
    double breadth;    // Breadth of a box
    double height;     // Height of a box
};

// Initialize static member of class Box
int Box::objectCount = 0;

int main(void)
{

  // Print total number of objects before creating object.
  cout << "Inital Stage Count: " << Box::getCount() << endl;
```

```
  Box Box1(3.3, 1.2, 1.5);    // Declare box1
  Box Box2(8.5, 6.0, 2.0);    // Declare box2

  // Print total number of objects after creating object.
  cout << "Final Stage Count: " << Box::getCount() << endl;

  return 0;
}
```

When the above code is compiled and executed, it produces following result:

```
Inital Stage Count: 0
Constructor called.
Constructor called.
Final Stage Count: 2
```

Mutable Data Members

Like static keyword C++ also supports the storage class specifier of mutable.

What is the need of mutable?
Sometimes there is requirement to modify one or more data members of class / struct through const function even though you don't want the function to update other members of class / struct. This task can be easily performed by using mutable keyword.

Consider this example where use of mutable can be useful. Suppose you go to hotel and you give the order to waiter to bring some food dish. After giving order, you suddenly decide to change the order of food. Assume that hotel provides facility to change the ordered food and again take the order of new food within 10 minutes after giving the 1st order. After 10 minutes order can't be cancelled and old order can't be replaced by new order. See the following code for details.

```
#include <iostream>
#include <string.h>
using std::cout;
using std::endl;

class Customer
{
    char name[25];
    mutable char placedorder[50];
    int tableno;
    mutable int bill;
public:
```

```
Customer(char* s, char* m, int a, int p)
    {
        strcpy(name, s);
        strcpy(placedorder, m);
        tableno = a;
        bill = p;
    }
    void changePlacedOrder(char* p) const
    {
        strcpy(placedorder, p);
    }
    void changeBill(int s) const
    {
        bill = s;
    }
    void display() const
    {
        cout << "Customer name is: " << name << endl;
        cout << "Food ordered by customer is: " << placedorder << endl;
        cout << "table no is: " << tableno << endl;
        cout << "Total payable amount: " << bill << endl;
    }
};
 int main()
{
    const Customer c1("Pravasi Meet", "Ice Cream", 3, 100);
    c1.display();
    c1.changePlacedOrder("GulabJammuns");
    c1.changeBill(150);
    c1.display();
    return 0;
}
```

Memory allocation with "new"

The following program is an example regarding the usage of memory allocation through "new" operator. For example, below program prints 10.

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

int main()

{
   int *n = new int(10); // initialization with new()

   cout<<*n;
   getchar();
   return 0;
}
```

Const Functions

A const member function guarantees that it will never modify any of its class's member data.

```
void conFunc( ) const
{
  x=10;

}
```

If there is separate declaration & definition of a function, const must be used both in declaration and definition. Member functions that do nothing but display object data are obvious members for being made const.

(Try creating const objects which access various member functions.)

Lab Tasks

Lab Task 1

Write a program to implement the Date class which has three data members named day, month and year. Provide two constructors, one which takes arguments and another one which does not (you may set values of data members to 0). Write getter and setter functions (input and show functions) for the class. Also write a member function CompareDates() which is called through one object and takes one object as parameter and tells which date comes first**.**

Example output:
13/11/2009 comes before 20/12/2009

Lab Task 2

Create a class "circle" having a data member radius (allocate memory with help of "new" operator in the constructor). The class should be capable of calculating area, and compare the areas of two circles displaying the area of which circle is greater. Make all the necessary function as needed including constructors. A=πr^2

Note:

Use const, mutable, static where required in your programs.

Deliverables

Submit source code of Task 1 and 2 on LMS for grading.