University American College Skopje

Course: Object Programming

Static members. Friends. This

Exercises

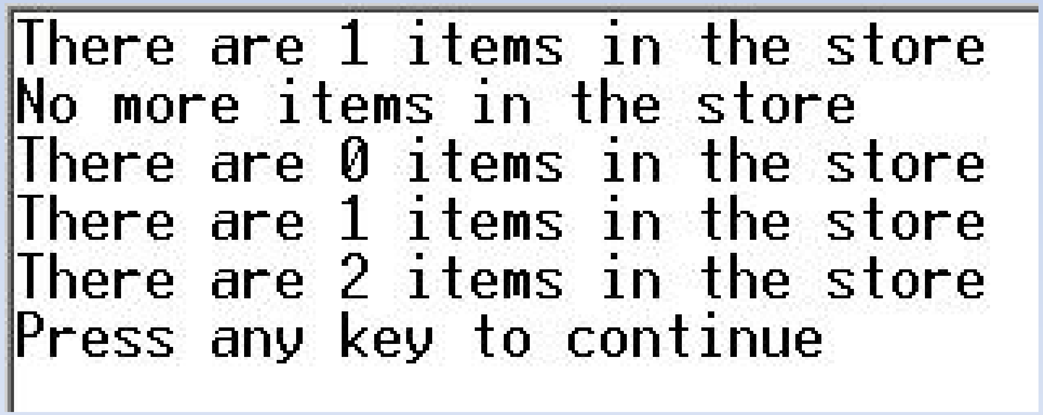
Prepared by: Biljana Stojčevska, PhD.
Edited for Spring semester 2013/14 by
Zlatko Ivanovski, M.Sc.

Assignment 1

- Create a class store which will have a private member int number of items and public members void show() (which will display the number of items) and a constructor which will initialize the number of items to zero
- Then create a class storekeeper which will be a friend of the class store. It will have two public members void add_items() and remove_items(), which will be able to access the number of items in the store and increase it and decrease it by one respectively.
- In the main() function, create a store and a storekeeper and demonstrate the effects of adding and removing items from the store

Assignment 1

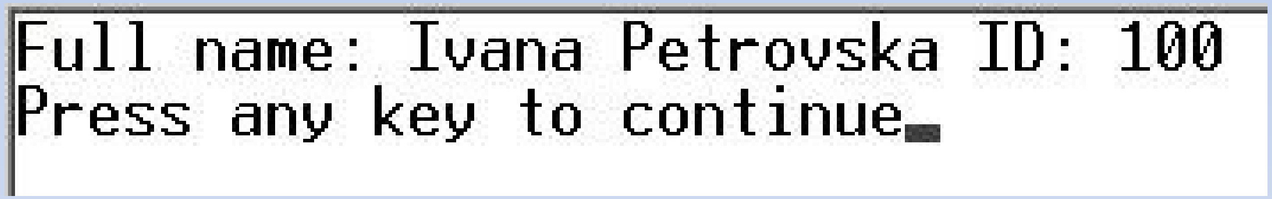
- The output:



```
There are 1 items in the store
No more items in the store
There are 0 items in the store
There are 1 items in the store
There are 2 items in the store
Press any key to continue
```

Assignment 2 (Assignment2.cpp)

- A demonstration for a friend class.



```
Full name: Ivana Petrovska ID: 100
Press any key to continue
```

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

```
class engineer;

class mathematician
{
public:
   void show(engineer); //show e member function of mathematician
private:
   int index;
   char fullname[20];
};
```

show is a member function of mathematician...

```
class engineer
{
public:
         engineer(char s[20], int i)
         {
                 strcpy(fullname, s);
                 id=i;
         }
         friend void mathematician::show(engineer);
private:
         int id;
         char fullname[20];
};
```

...but also a friend of the engineer class and thus has access to its non-public members

Definition of the function `show`

```
void mathematician::show(engineer i)
{
        cout<<"Full name: "<<i.fullname<<" ID: "<<i.id<<endl;
}
```

The `main()` function

```
void main()
{
        engineer I("Ivana Petrovska", 100);
        mathematician M;
        M.show(I);
}
```

Assignment 3 (Assignment3.cpp)

- Create a class Counter, which will have a private int member. It will have a public print() method and a public constructor to set the value of the private member to zero
- It will also have a friend global function setX(), which will be able to set the private member to a new value

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

```
class Counter {
   friend void setX( Counter &, int );
public:
   Counter() { x = 0; }
   void print()  { cout << x << endl; }
private:
   int x;
};
```

The `Counter` class has a `private` member, but the `setX` function has access to it, because it is declared as a `friend`

```
void setX( Counter &c, int val )
{
   c.x = val;
}
```

The definition of function `setX`

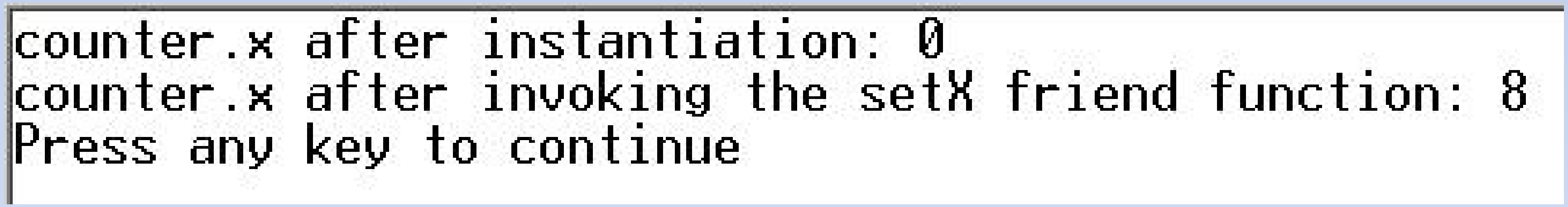
The `main()` function

```
int main()
{
   Counter counter;

   cout << "counter.x after instantiation: ";
   counter.print();
   cout << "counter.x after invoking the setX friend function: ";
   setX( counter, 8 );
   counter.print();
   return 0;
}
```

Assignment 3 (Assignment3.cpp)

- The output:



```
counter.x after instantiation: 0
counter.x after invoking the setX friend function: 8
Press any key to continue
```

Assignment 4

- Create a class Complex. It should have the following members:
  - two private double members real and imag
  - A public constructor with two parameters, to set the corresponding values of the private members
    - Make use of the **this** pointer upon initialization of the members
  - A public void print() function which will output the complex number in the form *real* + *imag**I
  - A public double abs() function which will return the absolute value of the complex number

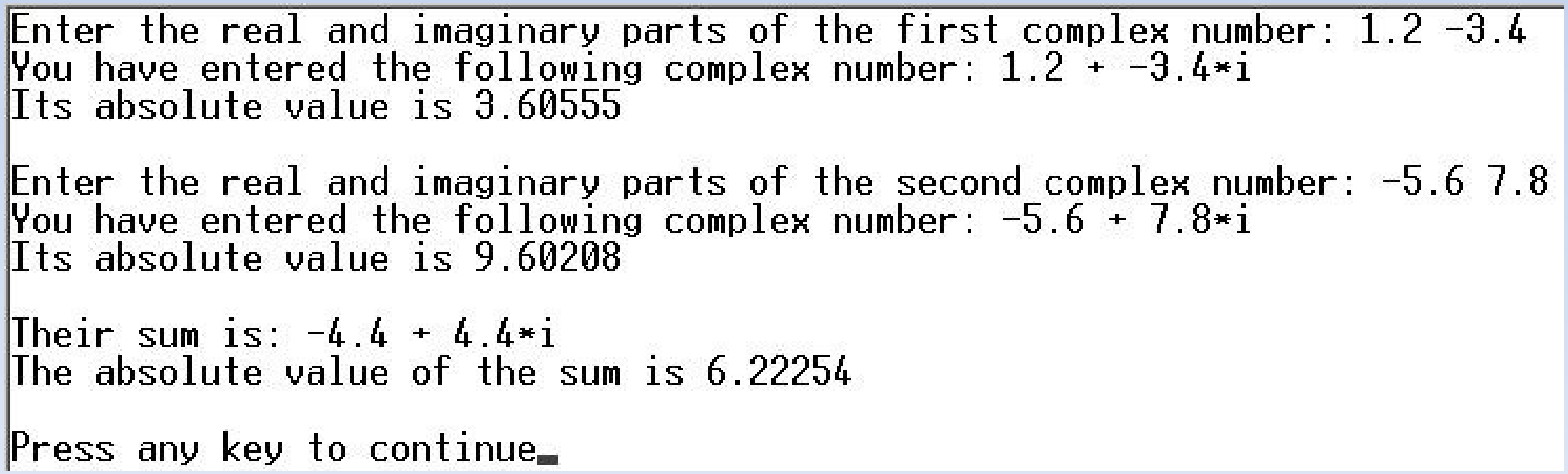
$$absolute\ value = \sqrt{(real * real) + (imag * imag)}$$

  - A friend double add() function, which will accept two Complex parameters, and return a Complex object, which will be the sum of the two Complex parameters

$$c3_{real} = c1_{real} + c2_{real}$$
$$c3_{imag} = c1_{imag} + c2_{imag}$$

Assignment 4

- In the main() function, create two Complex numbers, by giving the values for their real and imaginary parts through the keyboard. Upon creating them, print them on the screen and display their absolute values
- Then, using the add() function, obtain a third Complex number, which will be the sum of both Complex numbers. Also output it on the screen and show its absolute value



```
Enter the real and imaginary parts of the first complex number: 1.2 -3.4
You have entered the following complex number: 1.2 + -3.4*i
Its absolute value is 3.60555

Enter the real and imaginary parts of the second complex number: -5.6 7.8
You have entered the following complex number: -5.6 + 7.8*i
Its absolute value is 9.60208

Their sum is: -4.4 + 4.4*i
The absolute value of the sum is 6.22254

Press any key to continue_
```