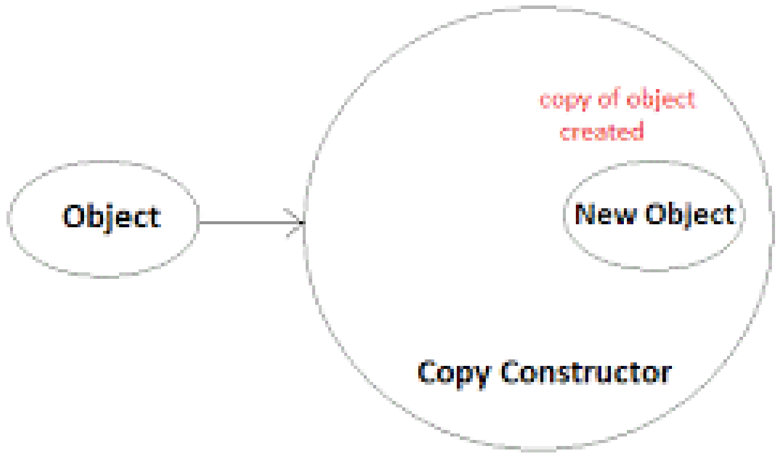
C/C++ Programming: C++ interm. (1/3)



Course **EE3093** – Lecturer: Dr F. Verdicchio

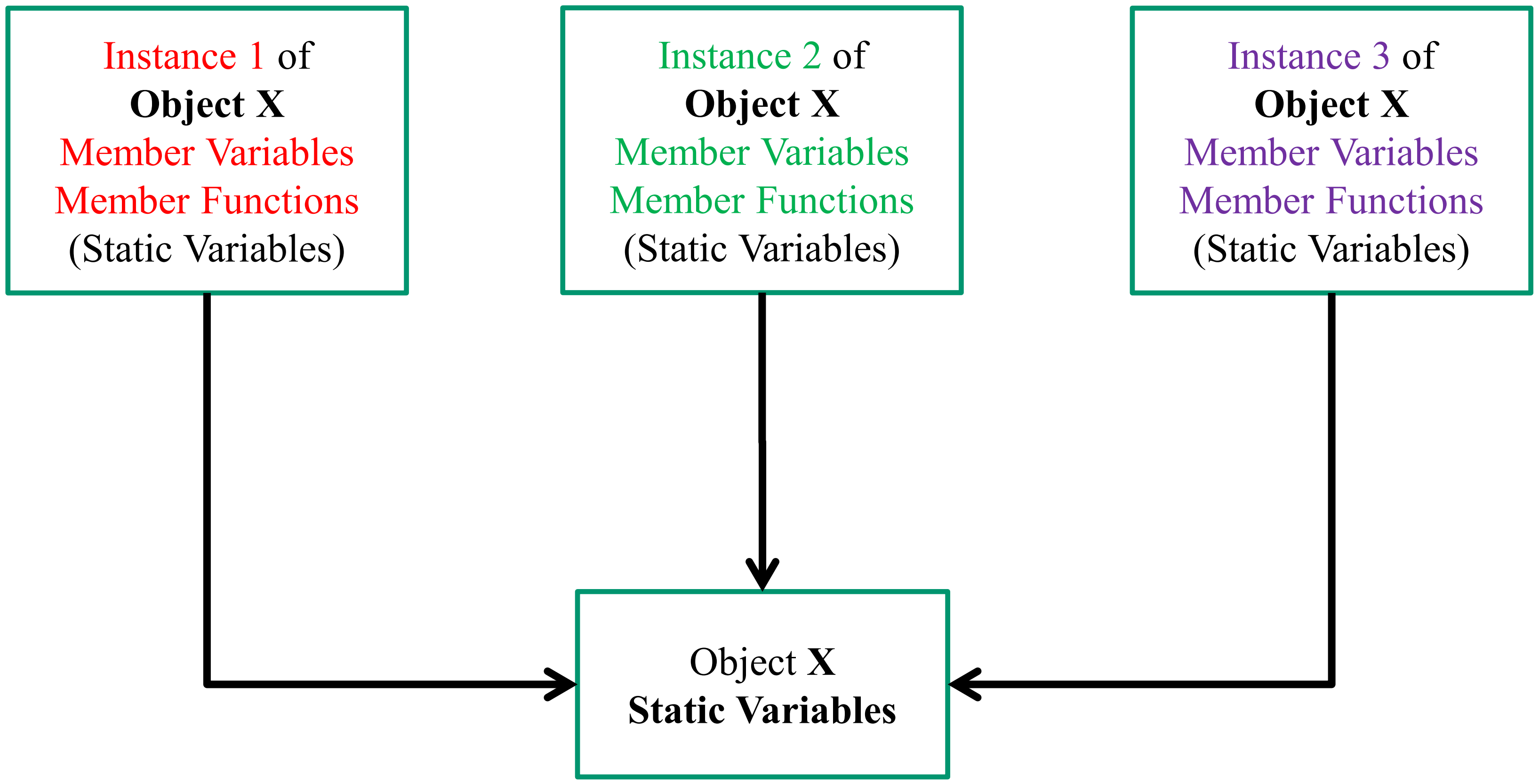
Any question?

Relevant topics to ask questions:

- C++ Objects:
  - constructors; composite objects; arrays of objects.
  - Projects; file inclusion; compilation; debugging.

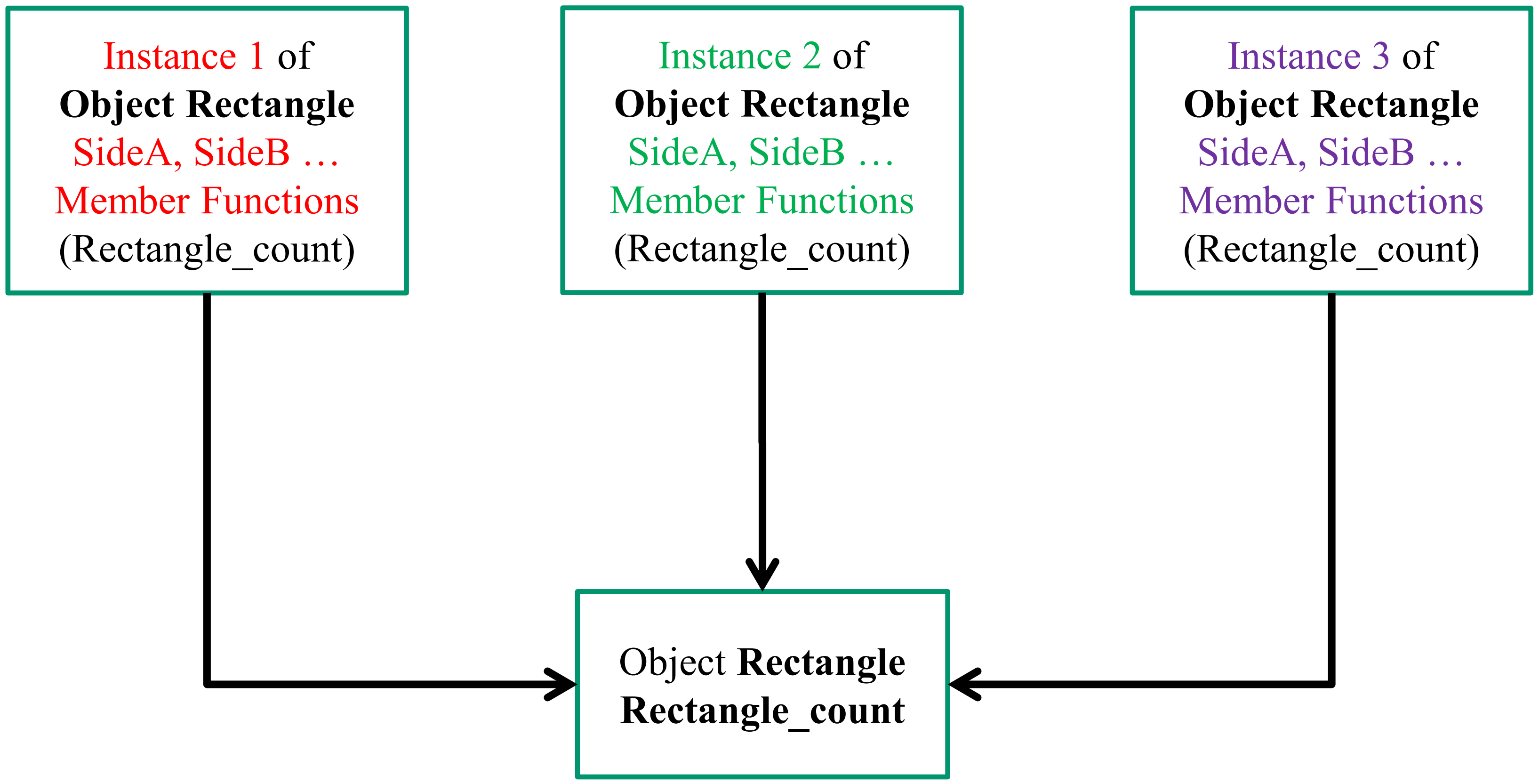
Static member variables

A **static member variable** is common across all instances of a class:

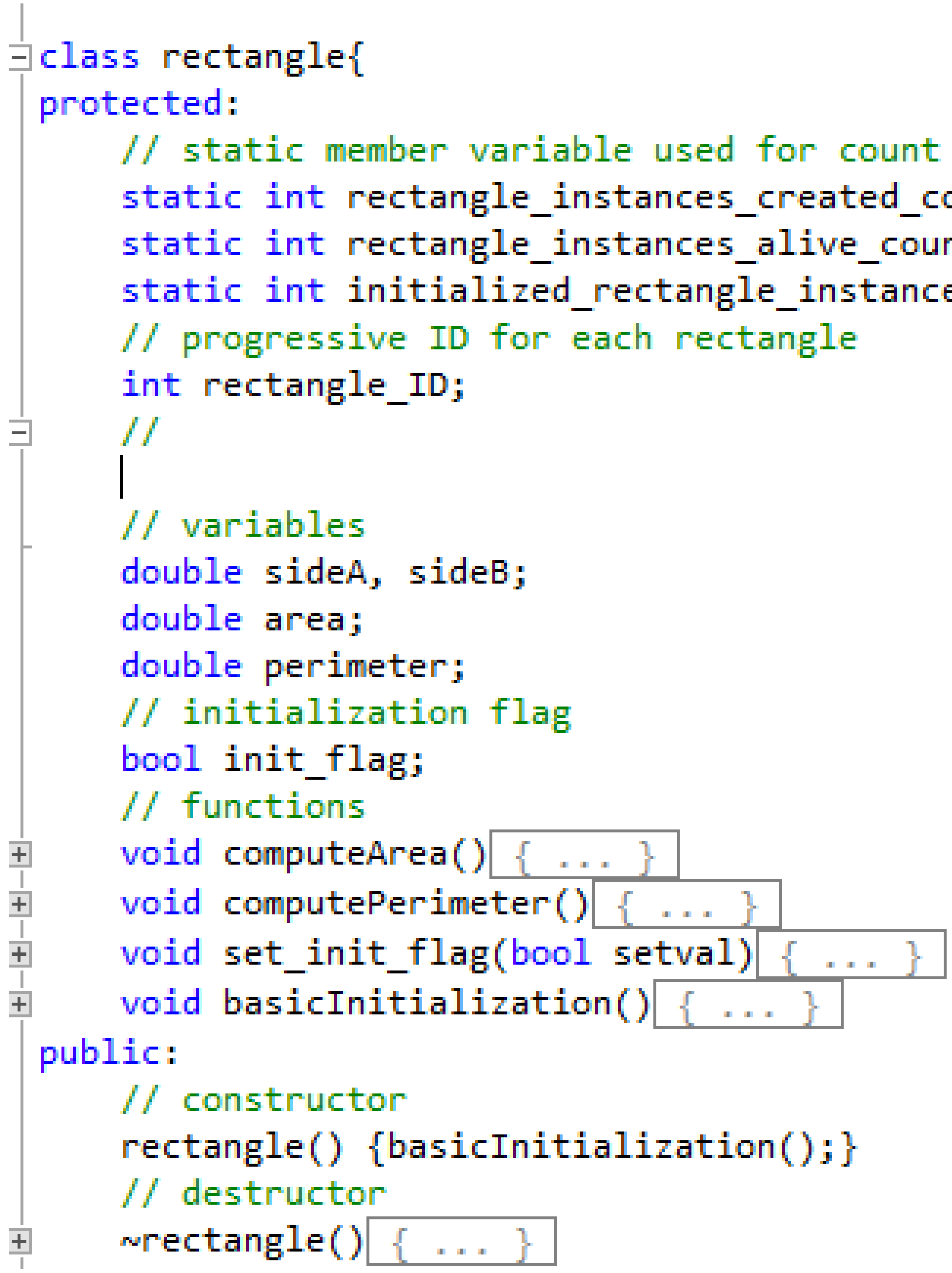


Static member variables

A **static member variable** is common across all instances of a class:



Static member variables: example

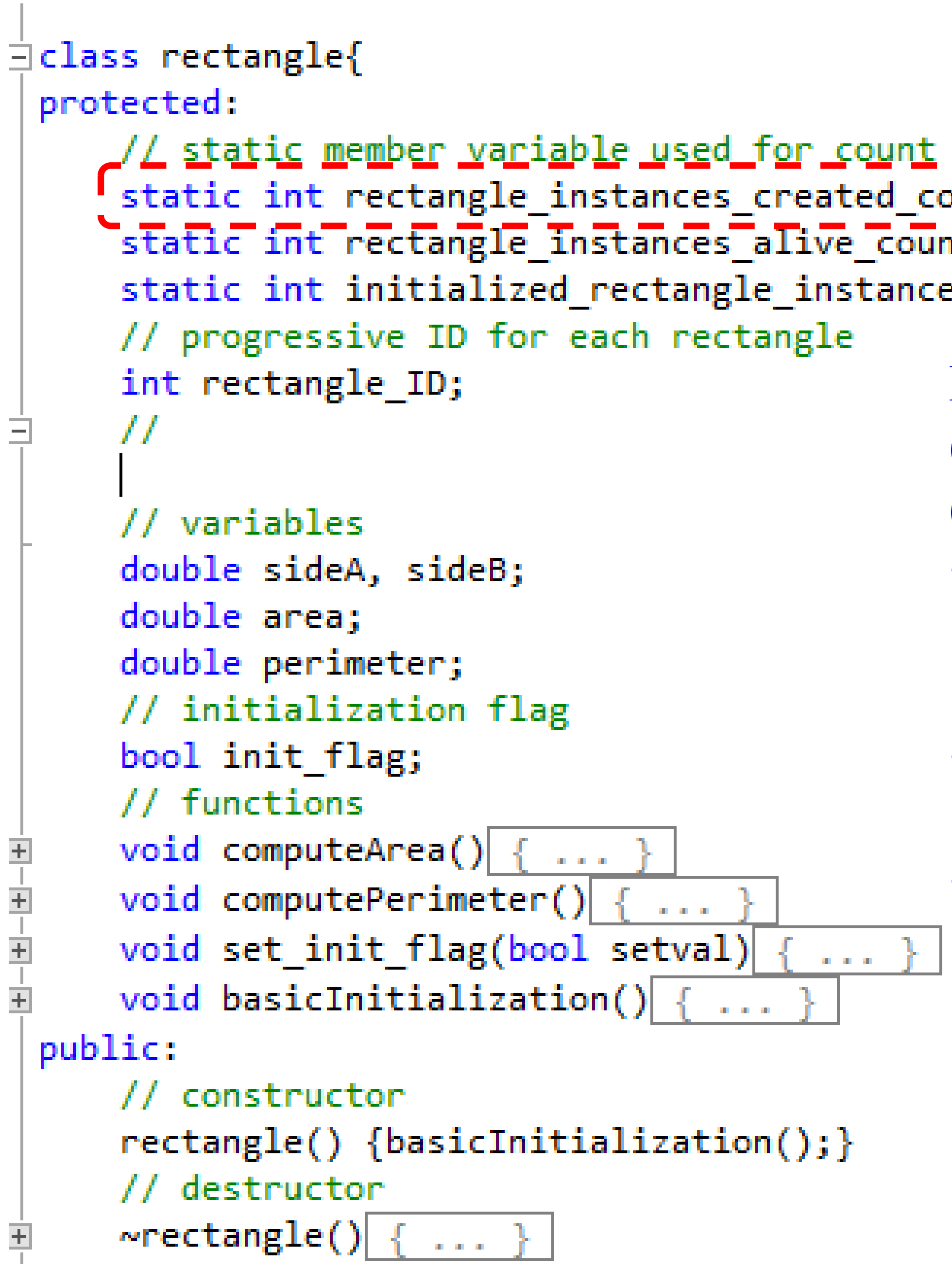


```
class rectangle{
protected:
    // static member variable used for count
    static int rectangle_instances_created_count;
    static int rectangle_instances_alive_count;
    static int initialized_rectangle_instances_count;
    // progressive ID for each rectangle
    int rectangle_ID;
    //
    
    // variables
    double sideA, sideB;
    double area;
    double perimeter;
    // initialization flag
    bool init_flag;
    // functions
    void computeArea() { ... }
    void computePerimeter() { ... }
    void set_init_flag(bool setval) { ... }
    void basicInitialization() { ... }
public:
    // constructor
    rectangle() {basicInitialization();}
    // destructor
    ~rectangle() { ... }
```

Keyword **static** used for these variables (type **int**) that are used to keep track (count) of:

- Instances of the class, i.e. Rectangle objects that have been instantiated (created) at some point;
- Instances of the class are still active (i.e. instantiated and not out of scope);
- Rectangles that have been initialized with valid input sideA, sideB

Static member variables: example

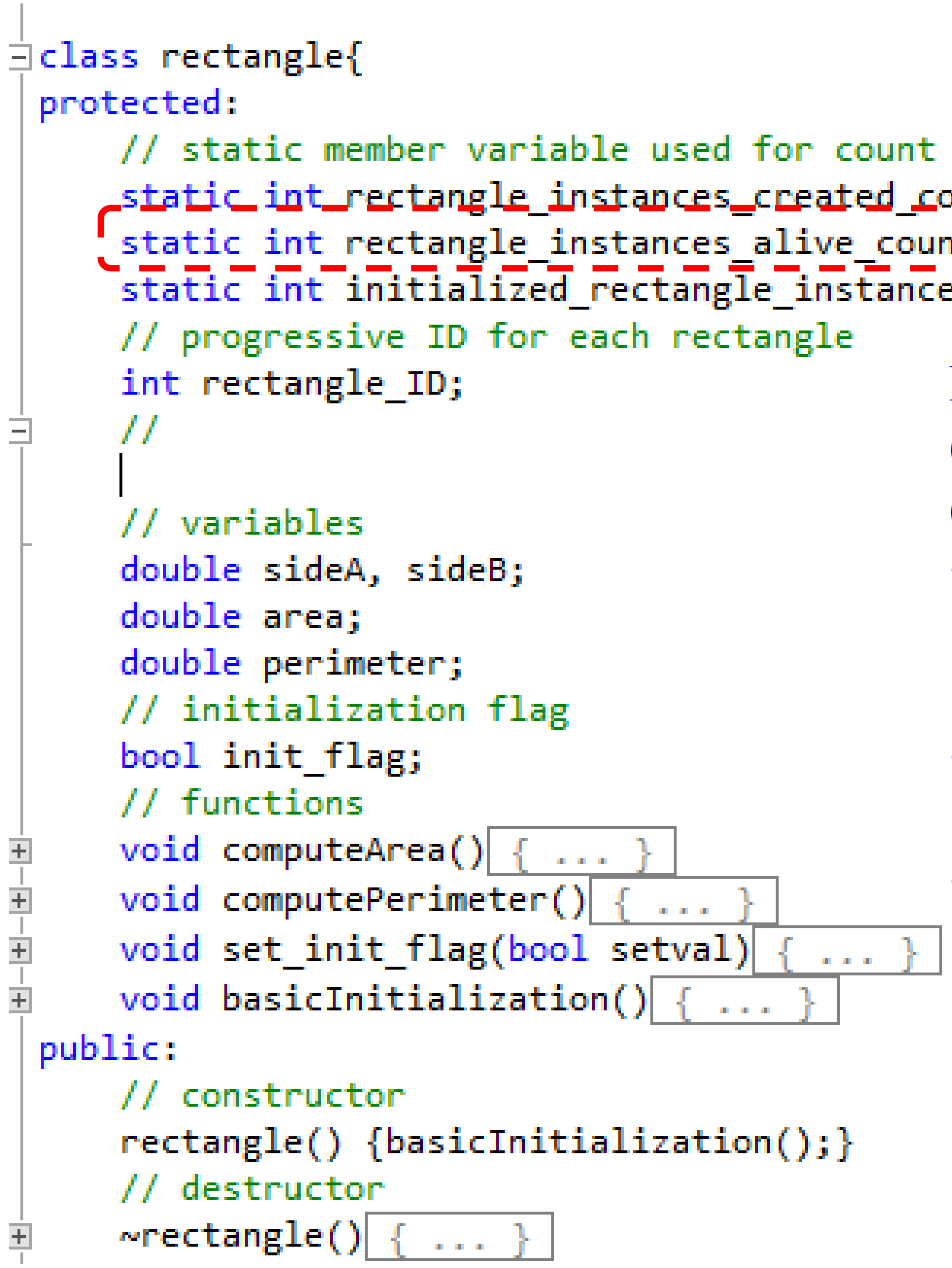


```
class rectangle{
protected:
    // static member variable used for count
    static int rectangle_instances_created_count;
    static int rectangle_instances_alive_count;
    static int initialized_rectangle_instances_count;
    // progressive ID for each rectangle
    int rectangle_ID;
    //

    // variables
    double sideA, sideB;
    double area;
    double perimeter;
    // initialization flag
    bool init_flag;
    // functions
    void computeArea() { ... }
    void computePerimeter() { ... }
    void set_init_flag(bool setval) { ... }
    void basicInitialization() { ... }
public:
    // constructor
    rectangle() {basicInitialization();}
    // destructor
    ~rectangle() { ... }
```

Keyword **static** used for these variables (type **int**) that are used to keep track (count) of:

- Instances of the class, i.e. Rectangle objects that have been instantiated (created) at some point;
- Instances of the class are still active (i.e. instantiated and not out of scope);
- Rectangles that have been initialized with valid input sideA, sideB

Static member variables: example

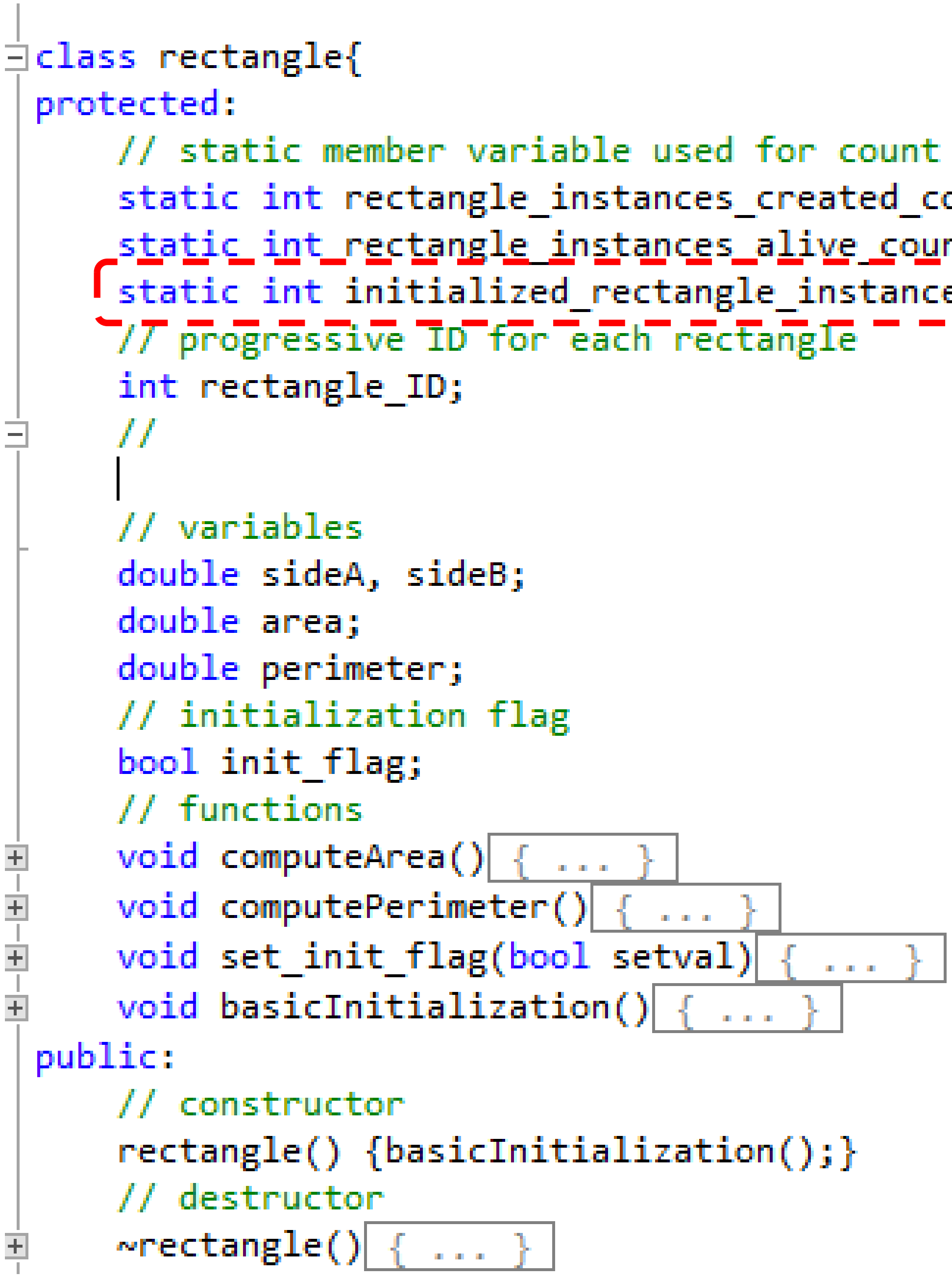


```
class rectangle{
protected:
    // static member variable used for count
    static int rectangle_instances_created_count;
    static int rectangle_instances_alive_count;
    static int initialized_rectangle_instances_count;
    // progressive ID for each rectangle
    int rectangle_ID;
    //

    // variables
    double sideA, sideB;
    double area;
    double perimeter;
    // initialization flag
    bool init_flag;
    // functions
    void computeArea() { ... }
    void computePerimeter() { ... }
    void set_init_flag(bool setval) { ... }
    void basicInitialization() { ... }
public:
    // constructor
    rectangle() {basicInitialization();}
    // destructor
    ~rectangle() { ... }
```

Keyword **static** used for these variables (type **int**) that are used to keep track (count) of:

- Instances of the class, i.e. Rectangle objects that have been instantiated (created) at some point;
- Instances of the class are still active (i.e. instantiated and not out of scope);
- Rectangles that have been initialized with valid input sideA, sideB

Static member variables: example

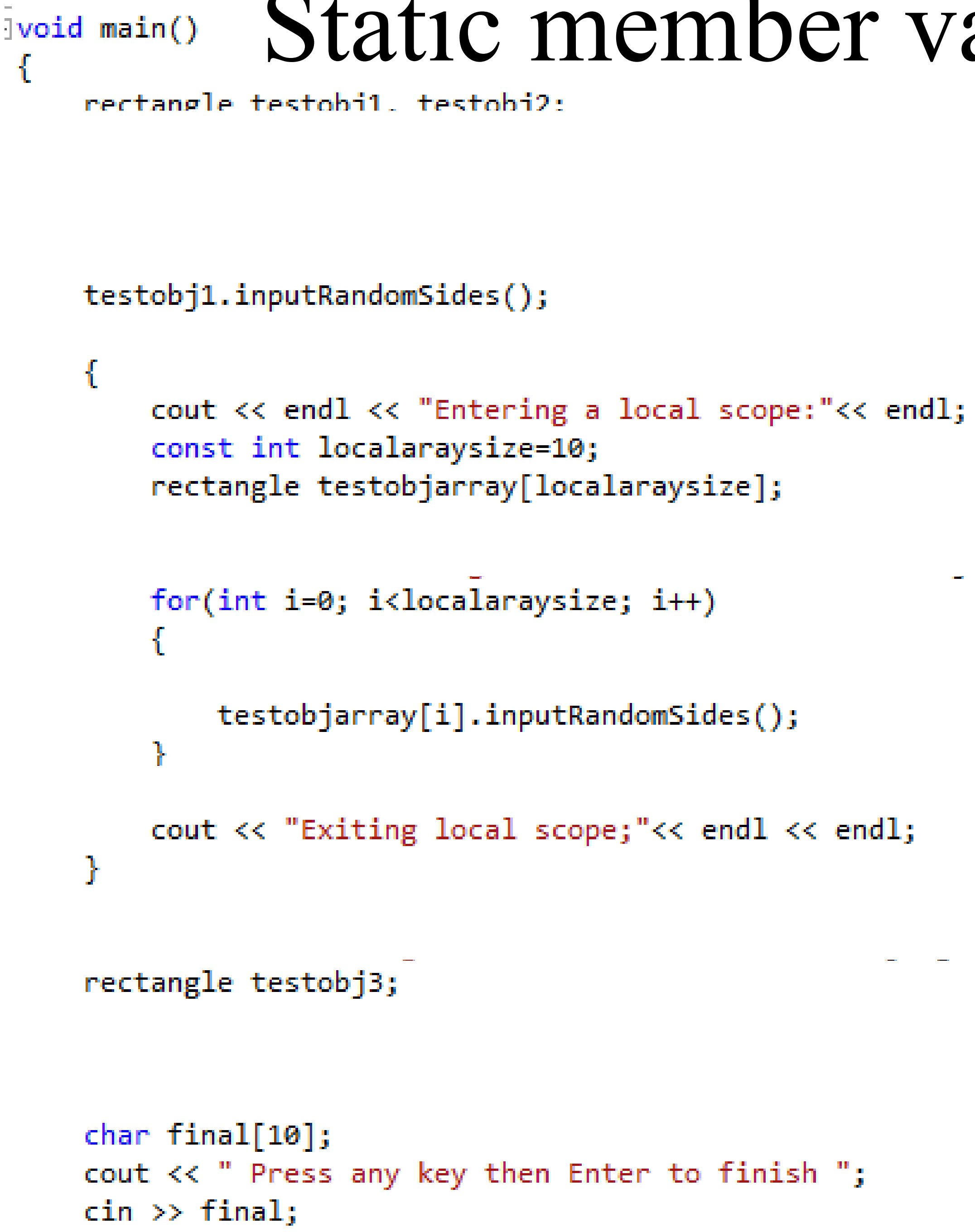


```
class rectangle{
protected:
    // static member variable used for count
    static int rectangle_instances_created_count;
    static int rectangle_instances_alive_count;
    static int initialized_rectangle_instances_count;
    // progressive ID for each rectangle
    int rectangle_ID;
    //
    |
    // variables
    double sideA, sideB;
    double area;
    double perimeter;
    // initialization flag
    bool init_flag;
    // functions
    void computeArea() { ... }
    void computePerimeter() { ... }
    void set_init_flag(bool setval) { ... }
    void basicInitialization() { ... }
public:
    // constructor
    rectangle() {basicInitialization();}
    // destructor
    ~rectangle() { ... }
```

Keyword **static** used for these variables (type **int**) that are used to keep track (count) of:

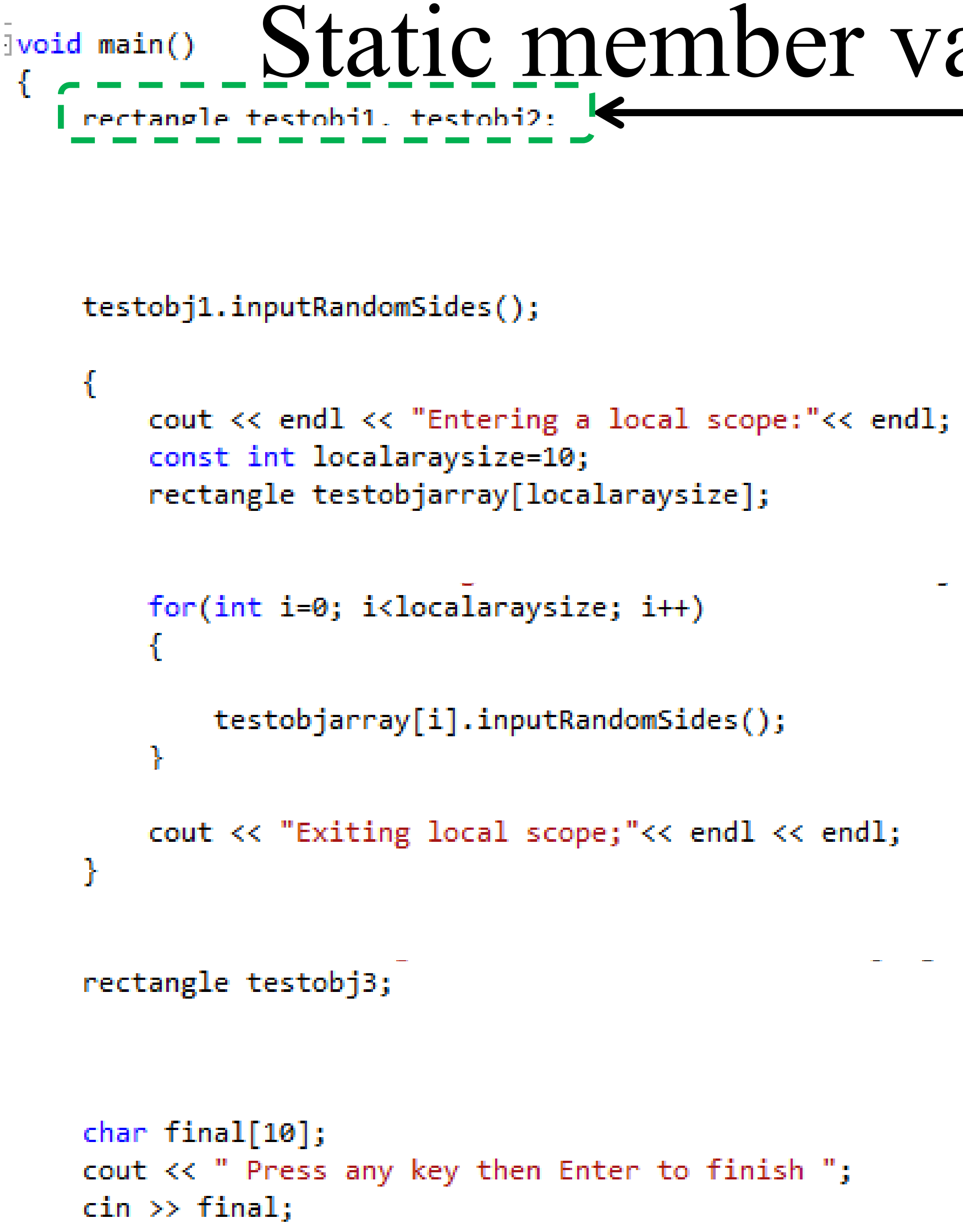
- Instances of the class, i.e. Rectangle objects that have been instantiated (created) at some point;
- Instances of the class are still active (i.e. instantiated and not out of scope);
- Rectangles that have been initialized with valid input sideA, sideB

Static member variables: use



```
void main()
{
    rectangle testobj1, testobj2;

    testobj1.inputRandomSides();

    {
        cout << endl << "Entering a local scope:"<< endl;
        const int localaraysize=10;
        rectangle testobjarray[localaraysize];

        for(int i=0; i<localaraysize; i++)
        {

            testobjarray[i].inputRandomSides();
        }

        cout << "Exiting local scope;"<< endl << endl;
    }

    rectangle testobj3;

    char final[10];
    cout << " Press any key then Enter to finish ";
    cin >> final;
```

Static member variables: use



```
void main()
{
    rectangle testobj1, testobj2;




    testobj1.inputRandomSides();

    {
        cout << endl << "Entering a local scope:"<< endl;
        const int localaraysize=10;
        rectangle testobjarray[localaraysize];


        for(int i=0; i<localaraysize; i++)
        {

            testobjarray[i].inputRandomSides();
        }

        cout << "Exiting local scope;"<< endl << endl;
    }


    rectangle testobj3;



    char final[10];
    cout << " Press any key then Enter to finish ";
    cin >> final;
```

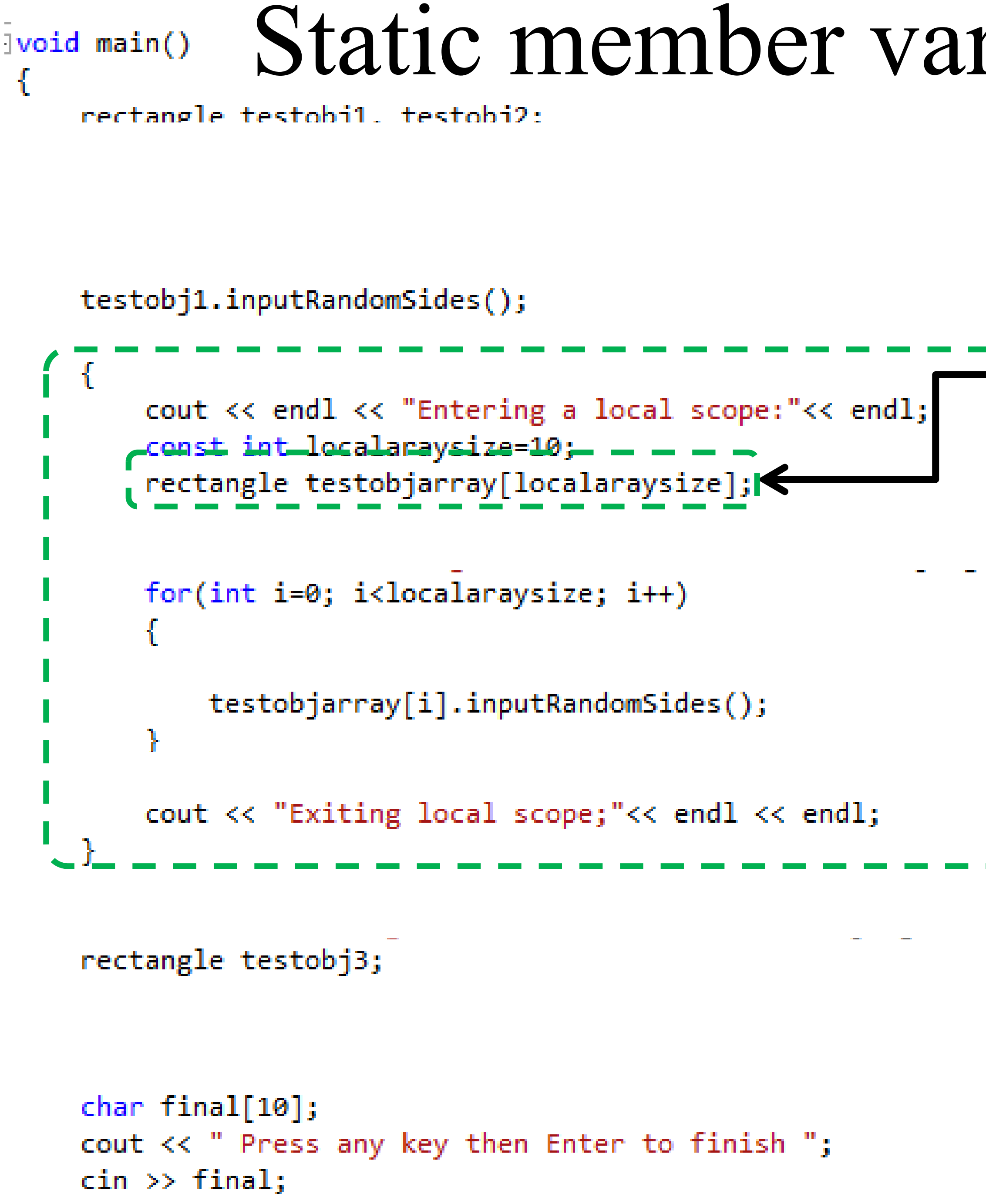
Two Instances of class Rectangle are instantiated (created);

Static member variables: use

```
void main()
{
    rectangle testobj1, testobj2;

    testobj1.inputRandomSides();

    {
        cout << endl << "Entering a local scope:"<< endl;
        const int localaraysize=10;
        rectangle testobjarray[localaraysize];

        for(int i=0; i<localaraysize; i++)
        {

            testobjarray[i].inputRandomSides();
        }

        cout << "Exiting local scope;"<< endl << endl;
    }

    rectangle testobj3;

    char final[10];
    cout << " Press any key then Enter to finish ";
    cin >> final;
```

One of them is initialized (with random values).

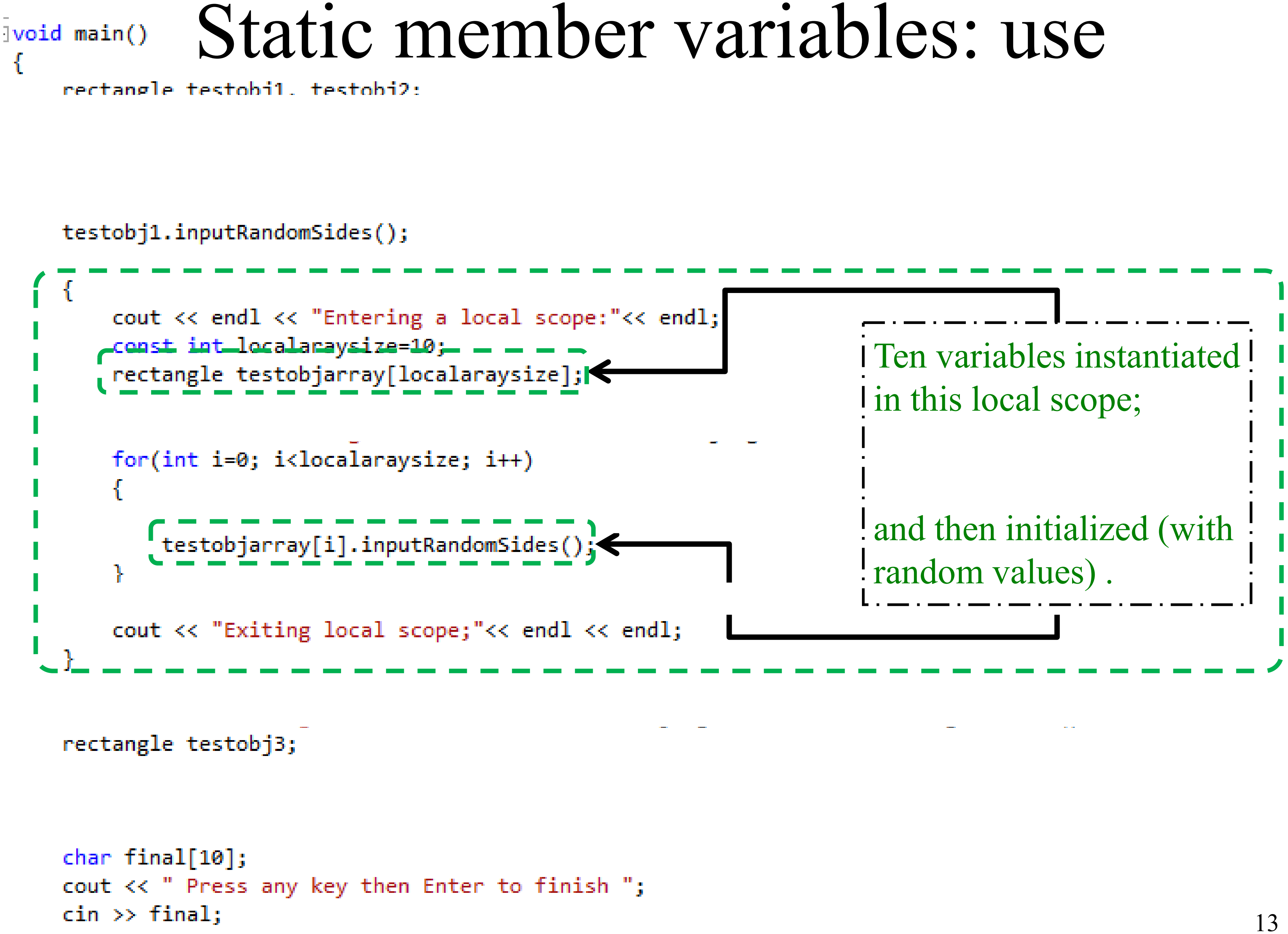
Static member variables: use



```
void main()
{
    rectangle testobj1, testobj2;




    testobj1.inputRandomSides();

    {
        cout << endl << "Entering a local scope:"<< endl;
        const int localaraysize=10;
        rectangle testobjarray[localaraysize];


        for(int i=0; i<localaraysize; i++)
        {

            testobjarray[i].inputRandomSides();
        }

        cout << "Exiting local scope;"<< endl << endl;
    }


    rectangle testobj3;




    char final[10];
    cout << " Press any key then Enter to finish ";
    cin >> final;
```

Ten variables instantiated in this local scope;

Static member variables: use

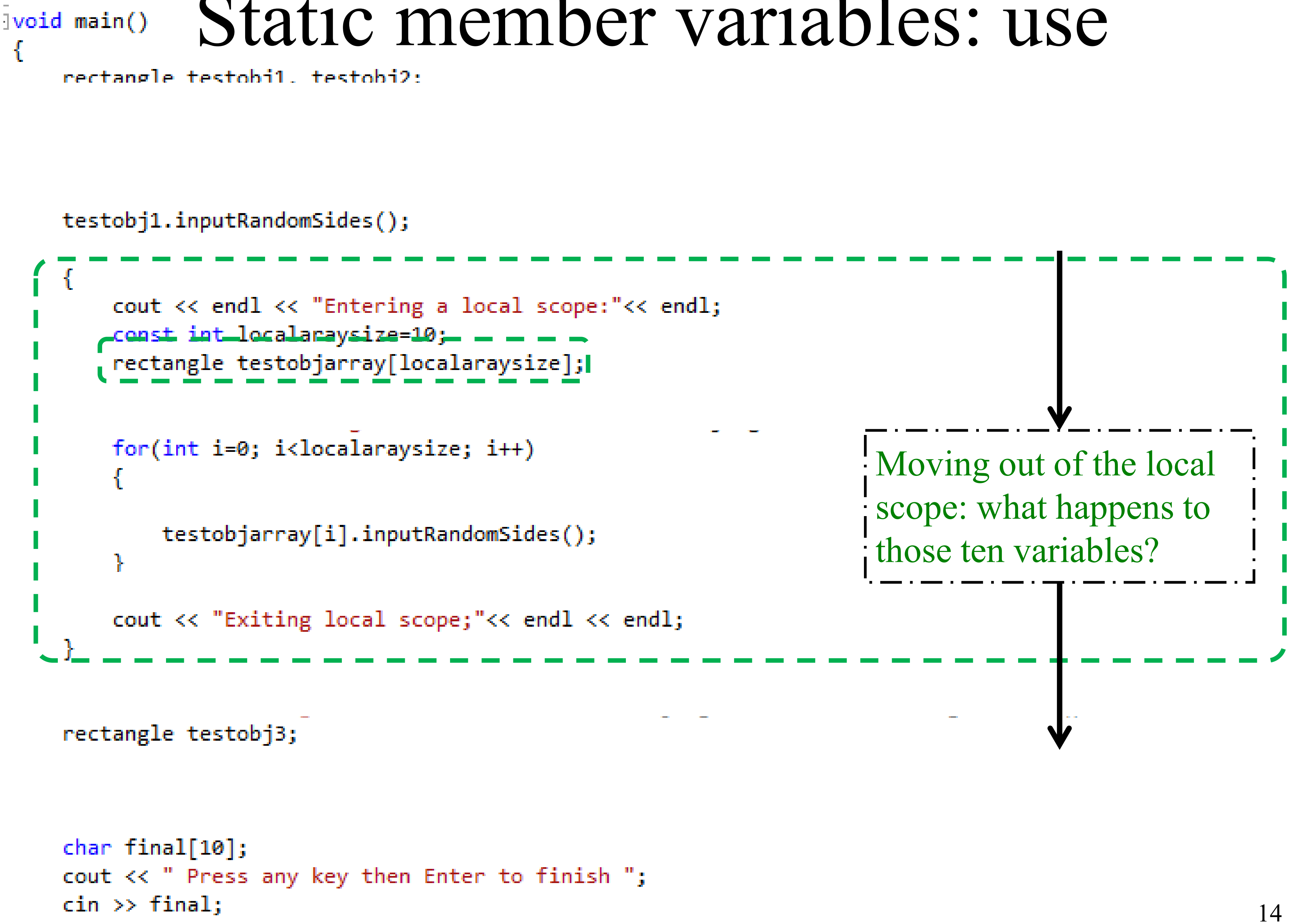


```
void main()
{
    rectangle testobj1, testobj2;

    testobj1.inputRandomSides();

    {
        cout << endl << "Entering a local scope:"<< endl;
        const int localaraysize=10;
        rectangle testobjarray[localaraysize];

        for(int i=0; i<localaraysize; i++)
        {
            testobjarray[i].inputRandomSides();
        }

        cout << "Exiting local scope;"<< endl << endl;
    }

    rectangle testobj3;

    char final[10];
    cout << " Press any key then Enter to finish ";
    cin >> final;
```

Ten variables instantiated in this local scope;

and then initialized (with random values) .

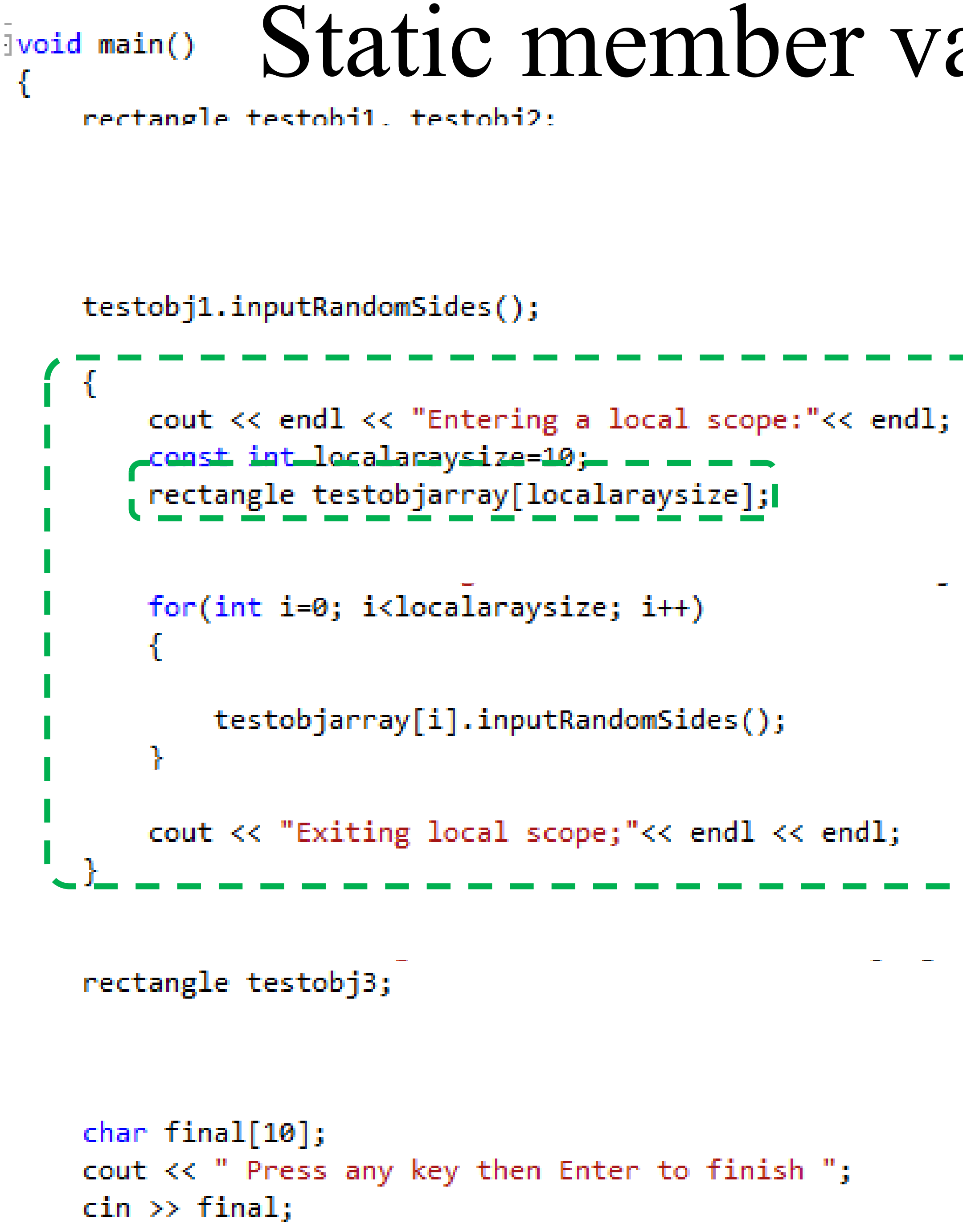
Static member variables: use



```
void main()
{
    rectangle testobj1, testobj2;

    testobj1.inputRandomSides();

    {
        cout << endl << "Entering a local scope:"<< endl;
        const int localaraysize=10;
        rectangle testobjarray[localaraysize];

        for(int i=0; i<localaraysize; i++)
        {

            testobjarray[i].inputRandomSides();
        }

        cout << "Exiting local scope;"<< endl << endl;
    }

    rectangle testobj3;

    char final[10];
    cout << " Press any key then Enter to finish ";
    cin >> final;
```

Moving out of the local scope: what happens to those ten variables?

Static member variables: use



```
void main()
{
    rectangle testobj1, testobj2;

    testobj1.inputRandomSides();

    {
        cout << endl << "Entering a local scope:"<< endl;
        const int localaraysize=10;
        rectangle testobjarray[localaraysize];

        for(int i=0; i<localaraysize; i++)
        {

            testobjarray[i].inputRandomSides();
        }

        cout << "Exiting local scope;"<< endl << endl;
    }

    rectangle testobj3;


    char final[10];
    cout << " Press any key then Enter to finish ";
    cin >> final;
```

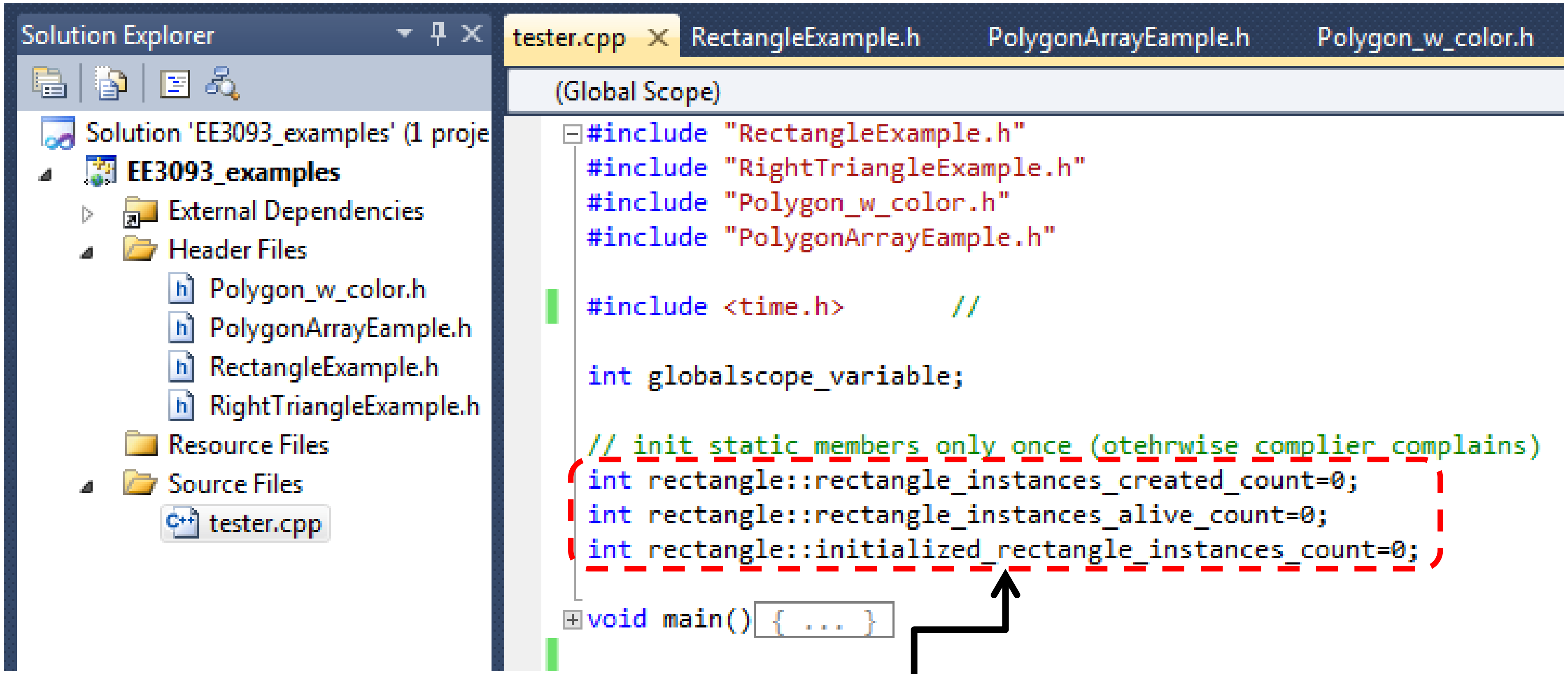
They go *out of scope* and are “lost” or “destroyed”

Static member variables: use

```
void main()
{
    rectangle testobj1, testobj2;

    testobj1.inputRandomSides();

    {
        cout << endl << "Entering a local scope:"<< endl;
        const int localaraysize=10;
        rectangle testobjarray[localaraysize];

        for(int i=0; i<localaraysize; i++)
        {

            testobjarray[i].inputRandomSides();
        }

        cout << "Exiting local scope;"<< endl << endl;
    }

    rectangle testobj3;

    char final[10];
    cout << " Press any key then Enter to finish ";
    cin >> final;
```

Another variable is instantiated

Static member variables: “how to”



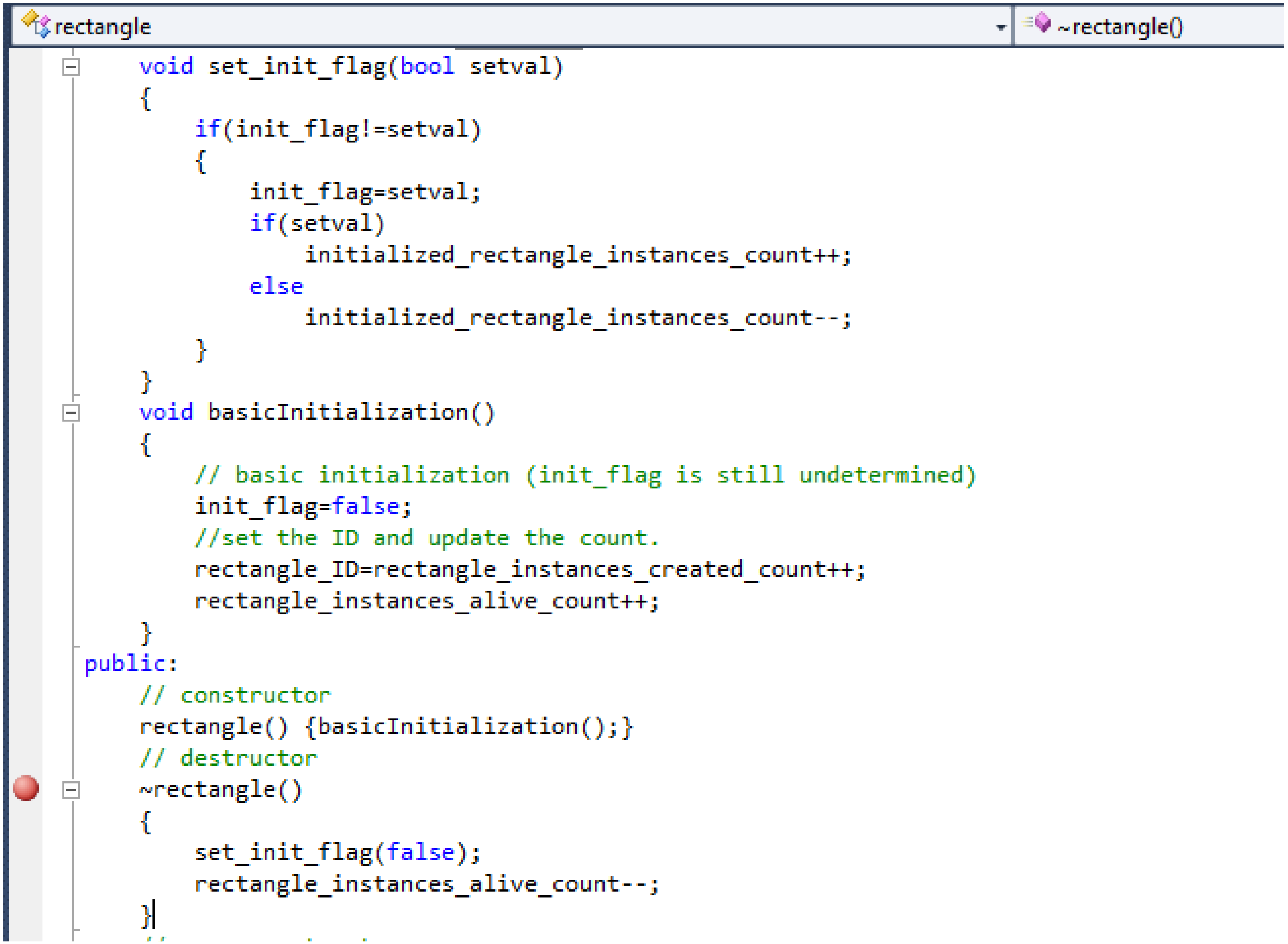
```
#include "RectangleExample.h"
#include "RightTriangleExample.h"
#include "Polygon_w_color.h"
#include "PolygonArrayEample.h"

#include <time.h>       //

int globalscope_variable;

// init static members only once (otehrwise complier complains)
int rectangle::rectangle_instances_created_count=0;
int rectangle::rectangle_instances_alive_count=0;
int rectangle::initialized_rectangle_instances_count=0;

void main() { ... }
```

Static member variables are:

- Instantiated (reporting the *type* and using the “ : : ” operator), i.e. the compiler allocates memory for those
- Initialized to an initial value (typically zero)

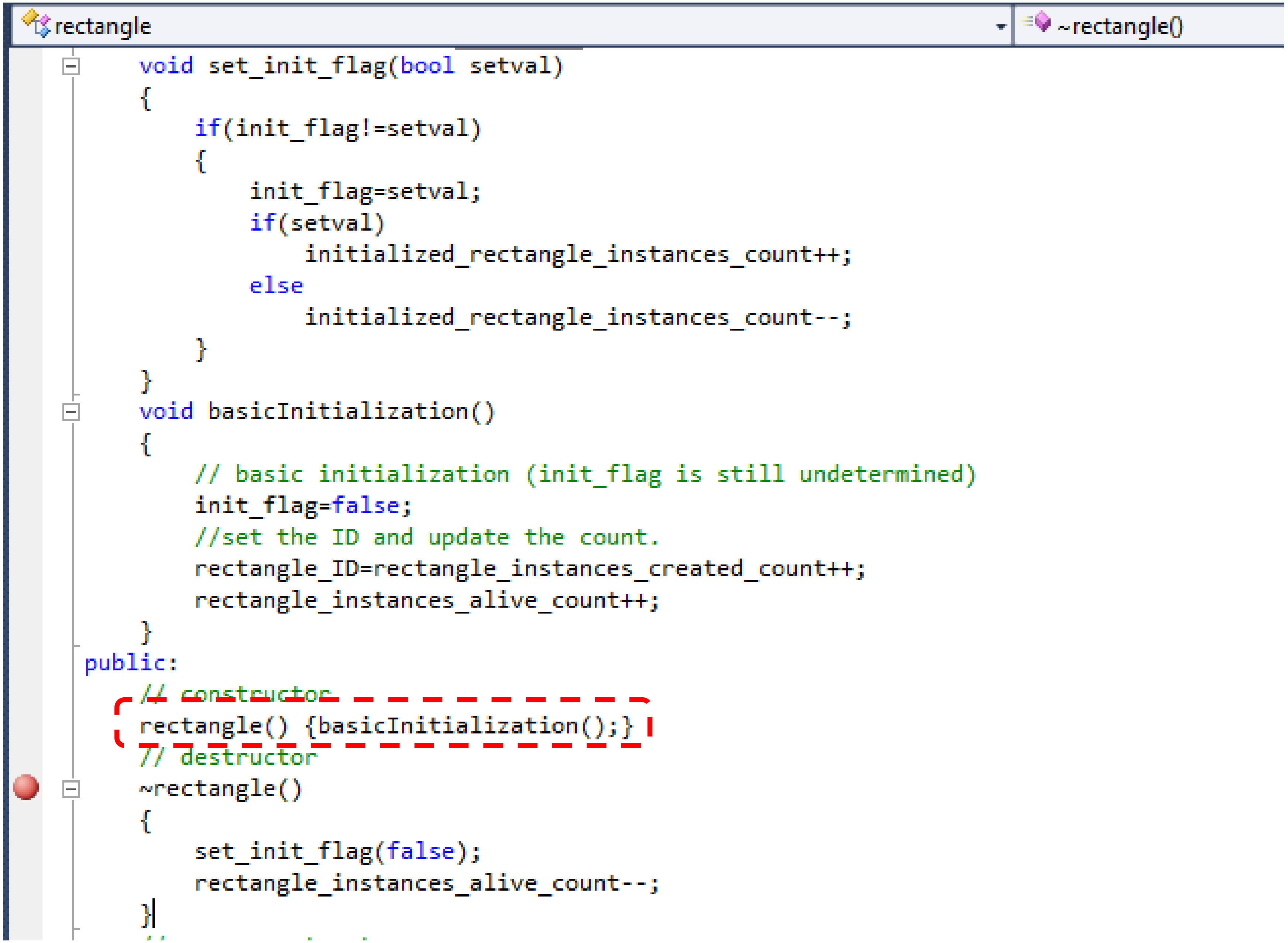
only **once** and at **global scope** (typically in a **cpp** file)

Static member variables: “how to”



```
    void set_init_flag(bool setval)
    {
        if(init_flag!=setval)
        {
            init_flag=setval;
            if(setval)
                initialized_rectangle_instances_count++;
            else
                initialized_rectangle_instances_count--;
        }
    }
    void basicInitialization()
    {
        // basic initialization (init_flag is still undetermined)
        init_flag=false;
        //set the ID and update the count.
        rectangle_ID=rectangle_instances_created_count++;
        rectangle_instances_alive_count++;
    }
public:
    // constructor
    rectangle() {basicInitialization();}
    // destructor
    ~rectangle()
    {
        set_init_flag(false);
        rectangle_instances_alive_count--;
    }
```

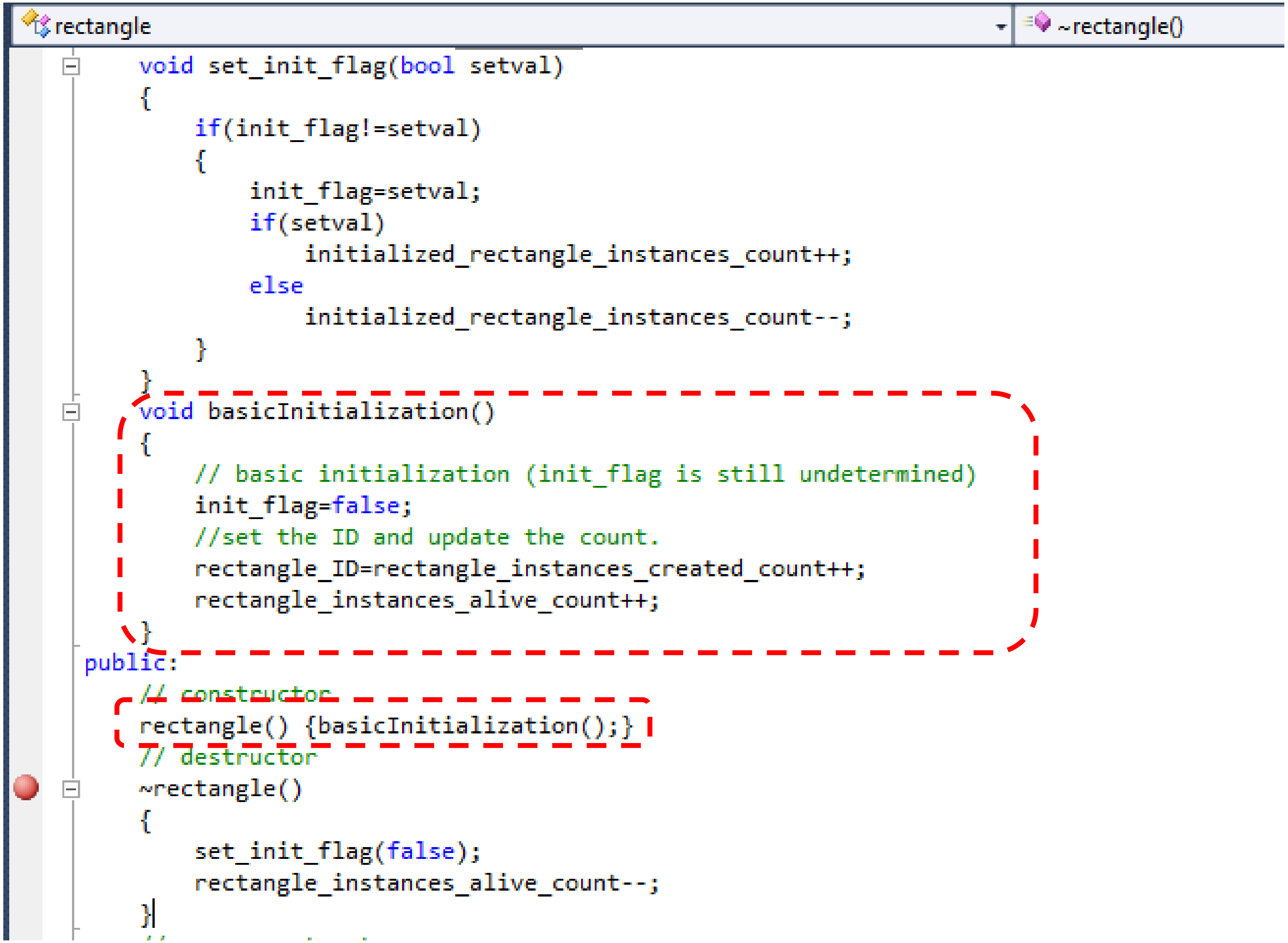
Static member variables: “how to”



rectangle ~rectangle()

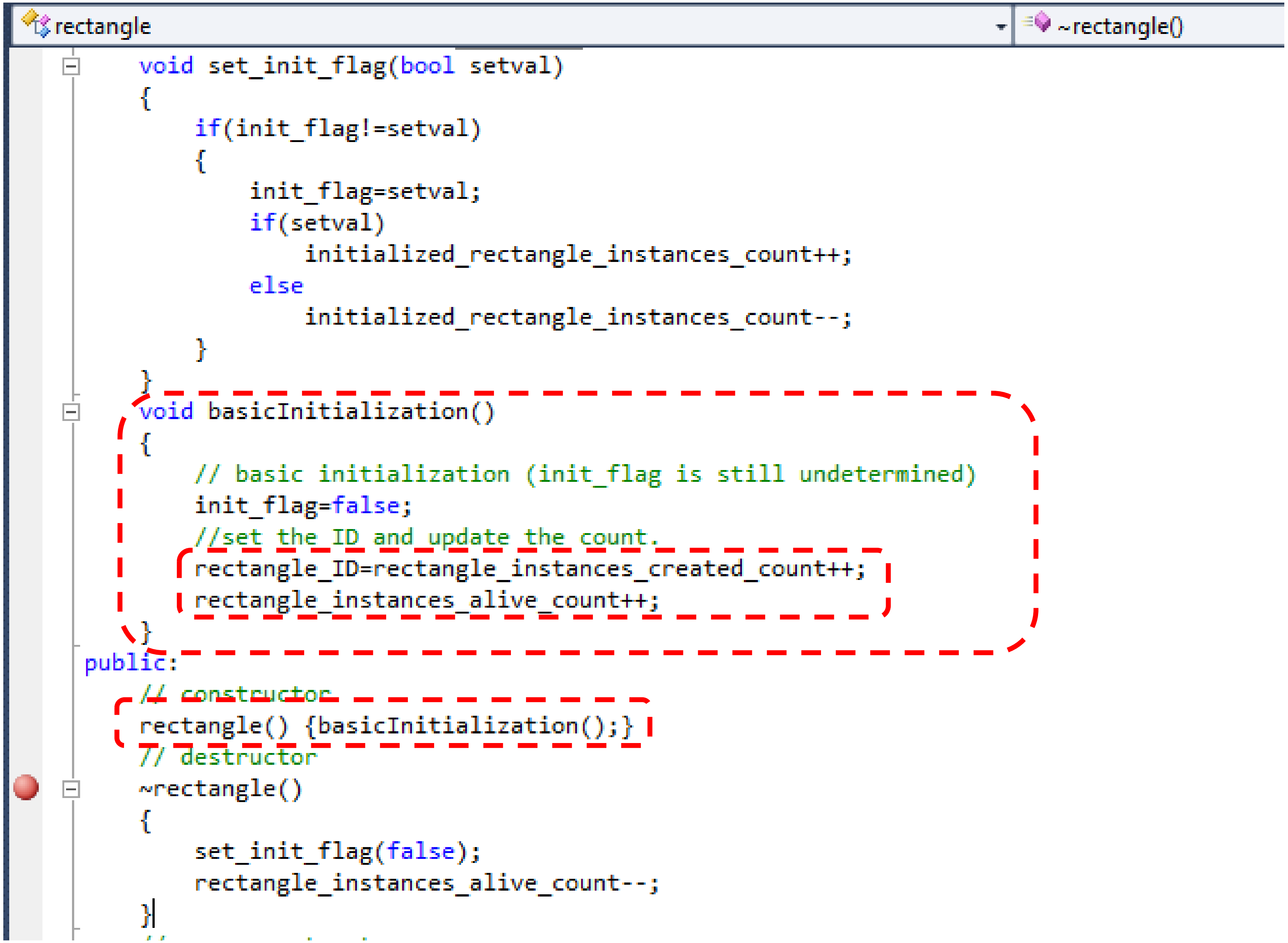
```
    void set_init_flag(bool setval)
    {
        if(init_flag!=setval)
        {
            init_flag=setval;
            if(setval)
                initialized_rectangle_instances_count++;
            else
                initialized_rectangle_instances_count--;
        }
    }
    void basicInitialization()
    {
        // basic initialization (init_flag is still undetermined)
        init_flag=false;
        //set the ID and update the count.
        rectangle_ID=rectangle_instances_created_count++;
        rectangle_instances_alive_count++;
    }
public:
    // constructor
    rectangle() {basicInitialization();}
    // destructor
    ~rectangle()
    {
        set_init_flag(false);
        rectangle_instances_alive_count--;
    }
```

Static member variables: “how to”



```
    void set_init_flag(bool setval)
    {
        if(init_flag!=setval)
        {
            init_flag=setval;
            if(setval)
                initialized_rectangle_instances_count++;
            else
                initialized_rectangle_instances_count--;
        }
    }
    void basicInitialization()
    {
        // basic initialization (init_flag is still undetermined)
        init_flag=false;
        //set the ID and update the count.
        rectangle_ID=rectangle_instances_created_count++;
        rectangle_instances_alive_count++;
    }
public:
    // constructor
    rectangle() {basicInitialization();}
    // destructor
    ~rectangle()
    {
        set_init_flag(false);
        rectangle_instances_alive_count--;
    }
```

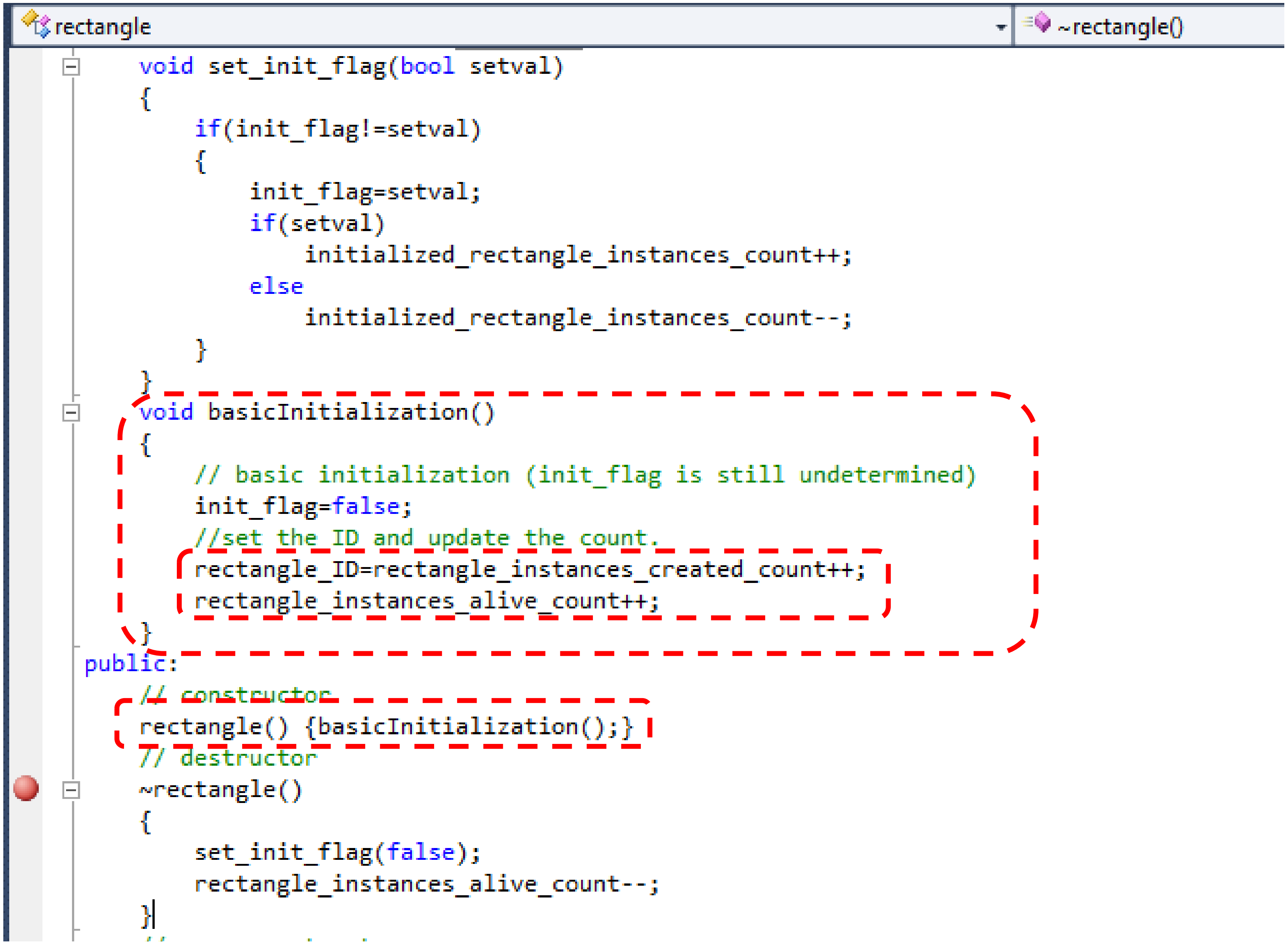
Static member variables: “how to”



rectangle ~rectangle()

```
    void set_init_flag(bool setval)
    {
        if(init_flag!=setval)
        {
            init_flag=setval;
            if(setval)
                initialized_rectangle_instances_count++;
            else
                initialized_rectangle_instances_count--;
        }
    }
    void basicInitialization()
    {
        // basic initialization (init_flag is still undetermined)
        init_flag=false;
        //set the ID and update the count.
        rectangle_ID=rectangle_instances_created_count++;
        rectangle_instances_alive_count++;
    }
public:
    // constructor
    rectangle() {basicInitialization();}
    // destructor
    ~rectangle()
    {
        set_init_flag(false);
        rectangle_instances_alive_count--;
    }
```

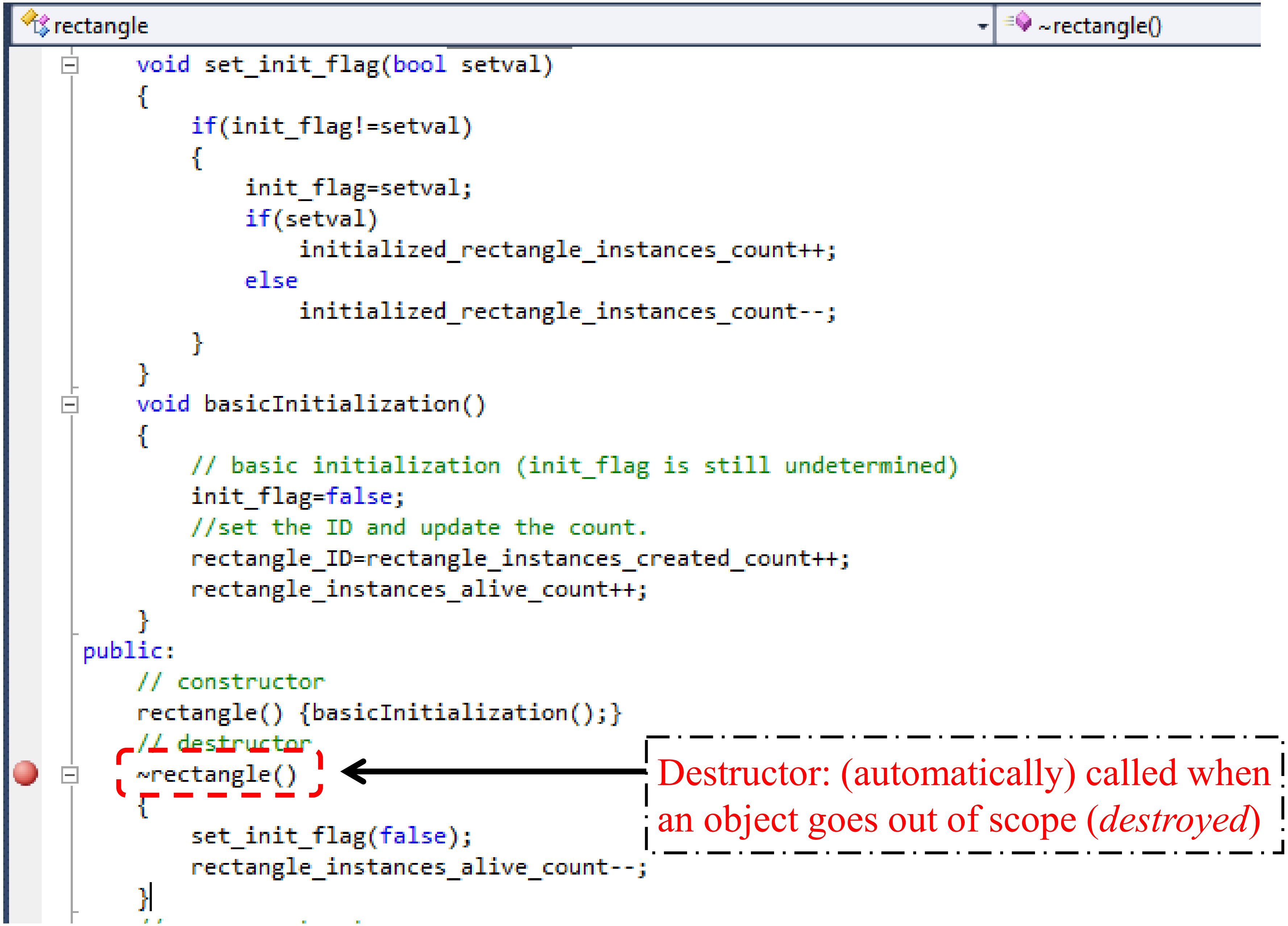
Static member variables: “how to”



rectangle ~rectangle()

```
    void set_init_flag(bool setval)
    {
        if(init_flag!=setval)
        {
            init_flag=setval;
            if(setval)
                initialized_rectangle_instances_count++;
            else
                initialized_rectangle_instances_count--;
        }
    }
    void basicInitialization()
    {
        // basic initialization (init_flag is still undetermined)
        init_flag=false;
        //set the ID and update the count.
        rectangle_ID=rectangle_instances_created_count++;
        rectangle_instances_alive_count++;
    }
public:
    // constructor
    rectangle() {basicInitialization();}
    // destructor
    ~rectangle()
    {
        set_init_flag(false);
        rectangle_instances_alive_count--;
    }
```

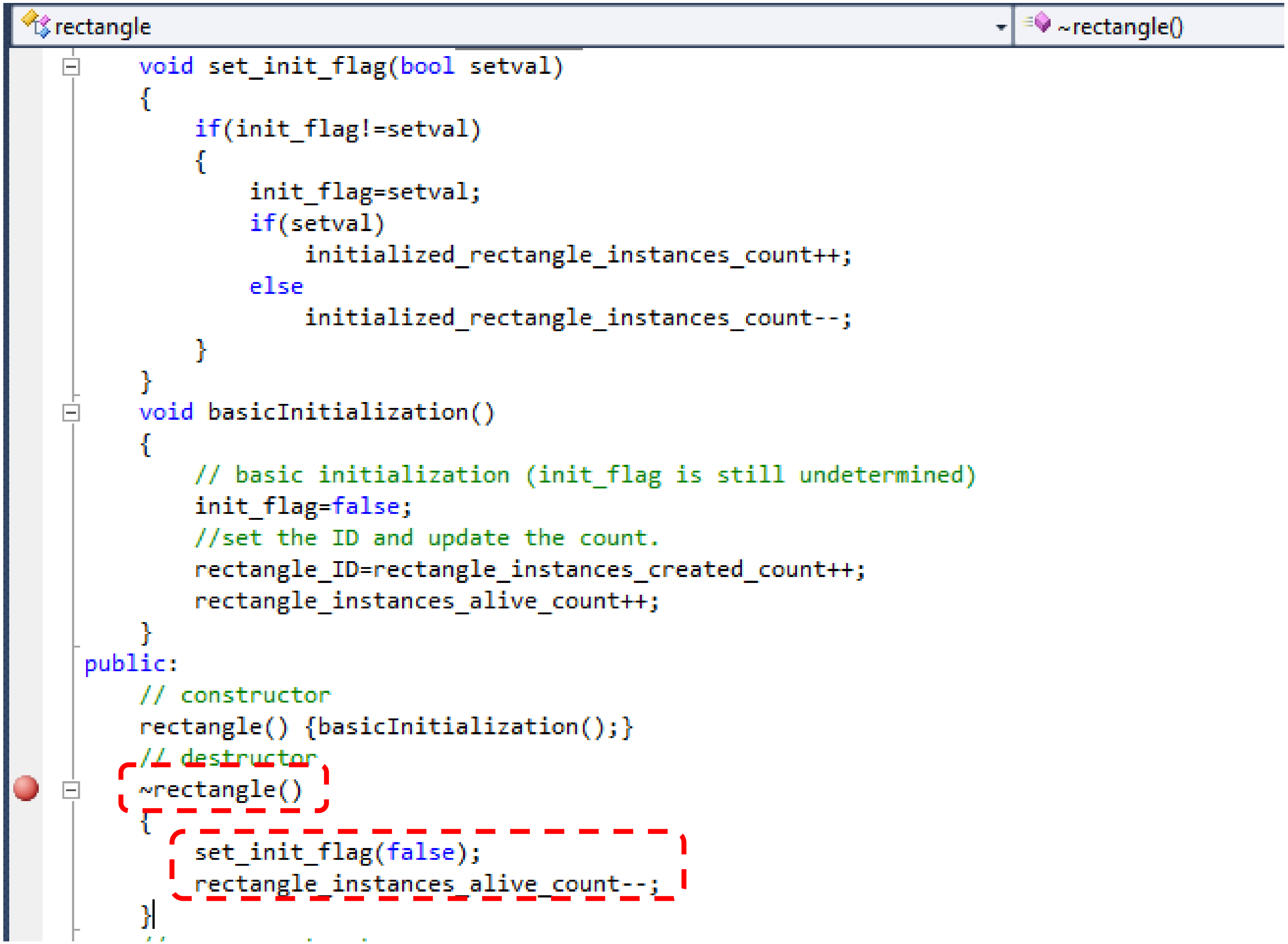
Static member variables: “how to”



```
    void set_init_flag(bool setval)
    {
        if(init_flag!=setval)
        {
            init_flag=setval;
            if(setval)
                initialized_rectangle_instances_count++;
            else
                initialized_rectangle_instances_count--;
        }
    }
    void basicInitialization()
    {
        // basic initialization (init_flag is still undetermined)
        init_flag=false;
        //set the ID and update the count.
        rectangle_ID=rectangle_instances_created_count++;
        rectangle_instances_alive_count++;
    }
public:
    // constructor
    rectangle() {basicInitialization();}
    // destructor
    ~rectangle()
    {
        set_init_flag(false);
        rectangle_instances_alive_count--;
    }
```

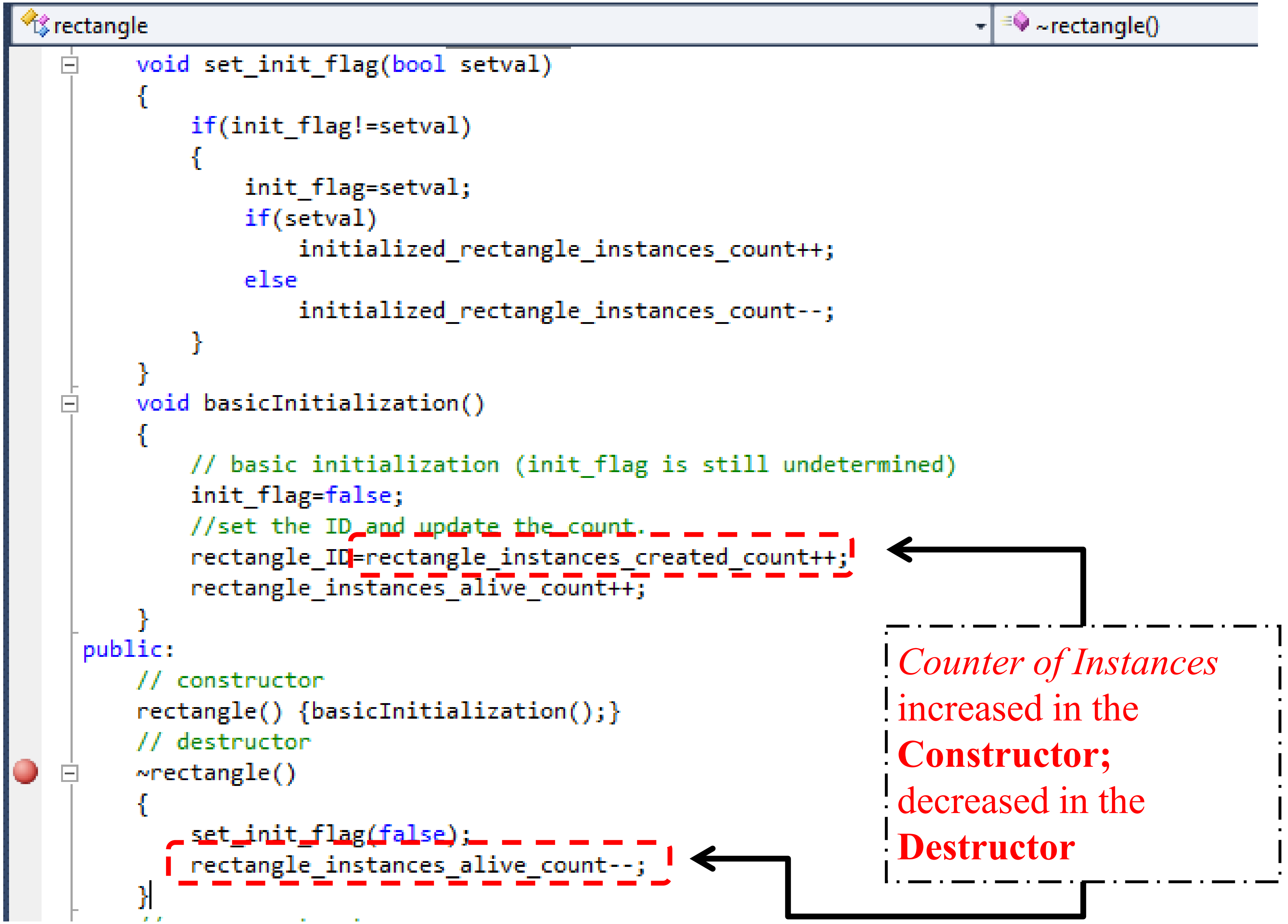
Destructor: (automatically) called when an object goes out of scope (*destroyed*)

Static member variables: “how to”



```
    void set_init_flag(bool setval)
    {
        if(init_flag!=setval)
        {
            init_flag=setval;
            if(setval)
                initialized_rectangle_instances_count++;
            else
                initialized_rectangle_instances_count--;
        }
    }
    void basicInitialization()
    {
        // basic initialization (init_flag is still undetermined)
        init_flag=false;
        //set the ID and update the count.
        rectangle_ID=rectangle_instances_created_count++;
        rectangle_instances_alive_count++;
    }
public:
    // constructor
    rectangle() {basicInitialization();}
    // destructor
    ~rectangle()
    {
        set_init_flag(false);
        rectangle_instances_alive_count--;
    }
```

Static member variables: “how to”



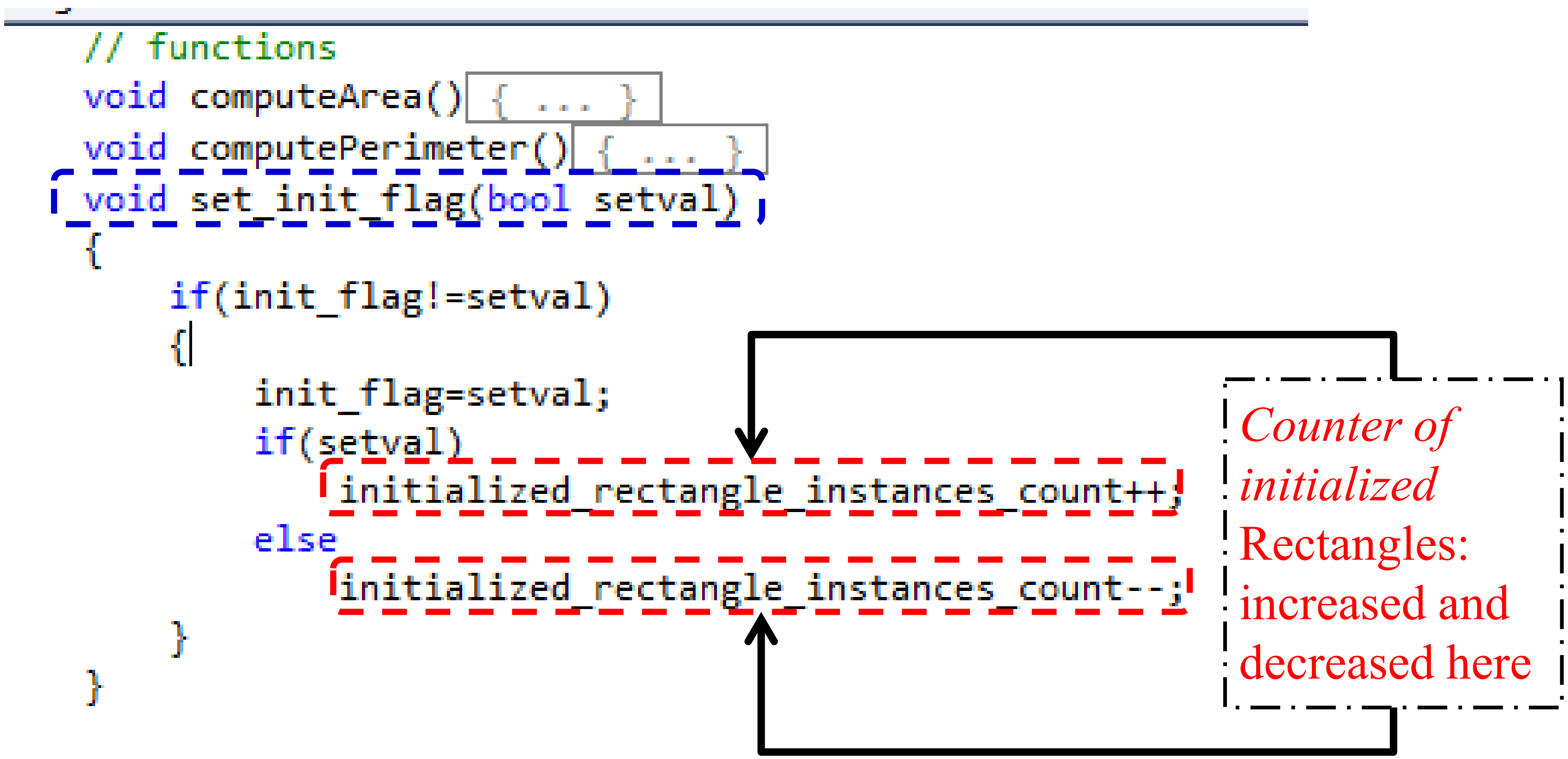
```
    void set_init_flag(bool setval)
    {
        if(init_flag!=setval)
        {
            init_flag=setval;
            if(setval)
                initialized_rectangle_instances_count++;
            else
                initialized_rectangle_instances_count--;
        }
    }
    void basicInitialization()
    {
        // basic initialization (init_flag is still undetermined)
        init_flag=false;
        //set the ID and update the count.
        rectangle_ID=rectangle_instances_created_count++;
        rectangle_instances_alive_count++;
    }
public:
    // constructor
    rectangle() {basicInitialization();}
    // destructor
    ~rectangle()
    {
        set_init_flag(false);
        rectangle_instances_alive_count--;
    }
```

Counter of Instances increased in the **Constructor;** decreased in the **Destructor**

Static member variables: “how to”

```
void inputSides(double in_sideA, double in_sideB)
{
    if(!isInitialized())
    {
        if(in_sideA>0 && in_sideB>0)
        {
            sideA=in_sideA;
            sideB=in_sideB;
            set_init_flag(true);
            computeArea();
            computePerimeter();
        }
        else
            cout << "Error in inputSides(): Incorrect input values" << endl;
    }
    else
        cout << "Error in inputSides(): Rectangle is already initialized " << endl;
}
```

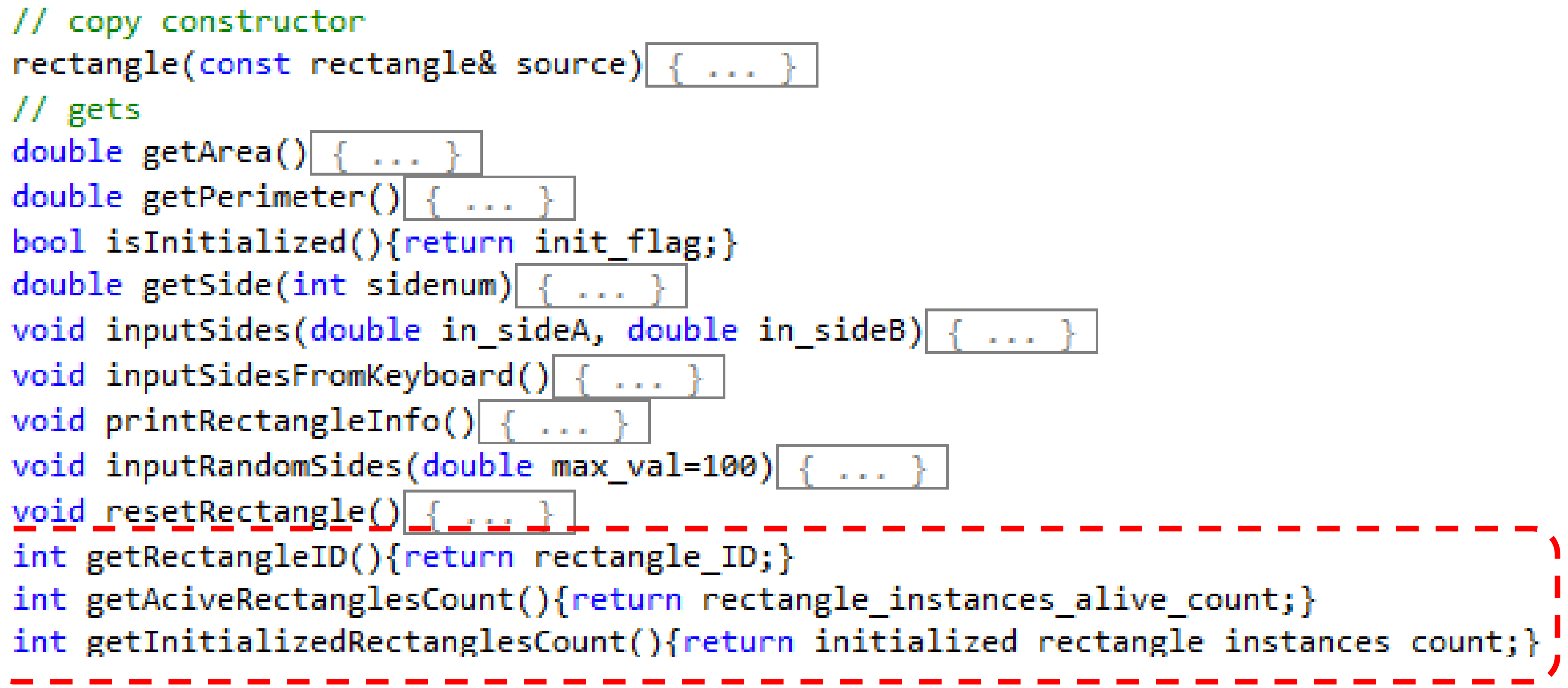
Static member variables: “how to”



```
    // functions
    void computeArea() { ... }
    void computePerimeter() { ... }
    void set_init_flag(bool setval)
    {
        if(init_flag!=setval)
        {
            init_flag=setval;
            if(setval)
                initialized_rectangle_instances_count++;
            else
                initialized_rectangle_instances_count--;
        }
    }
```

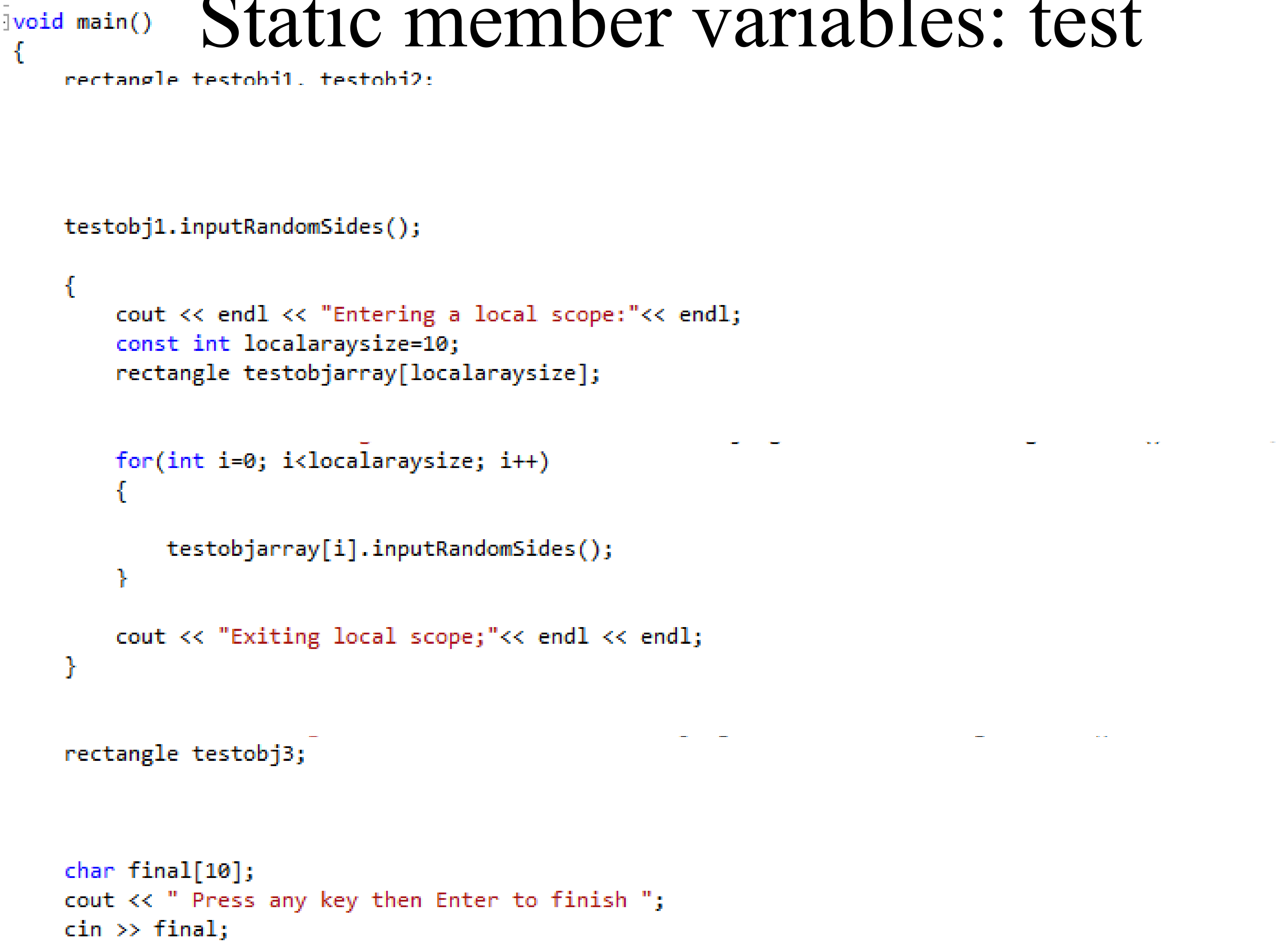
Counter of initialized Rectangles: increased and decreased here

Static member variables: “how to”



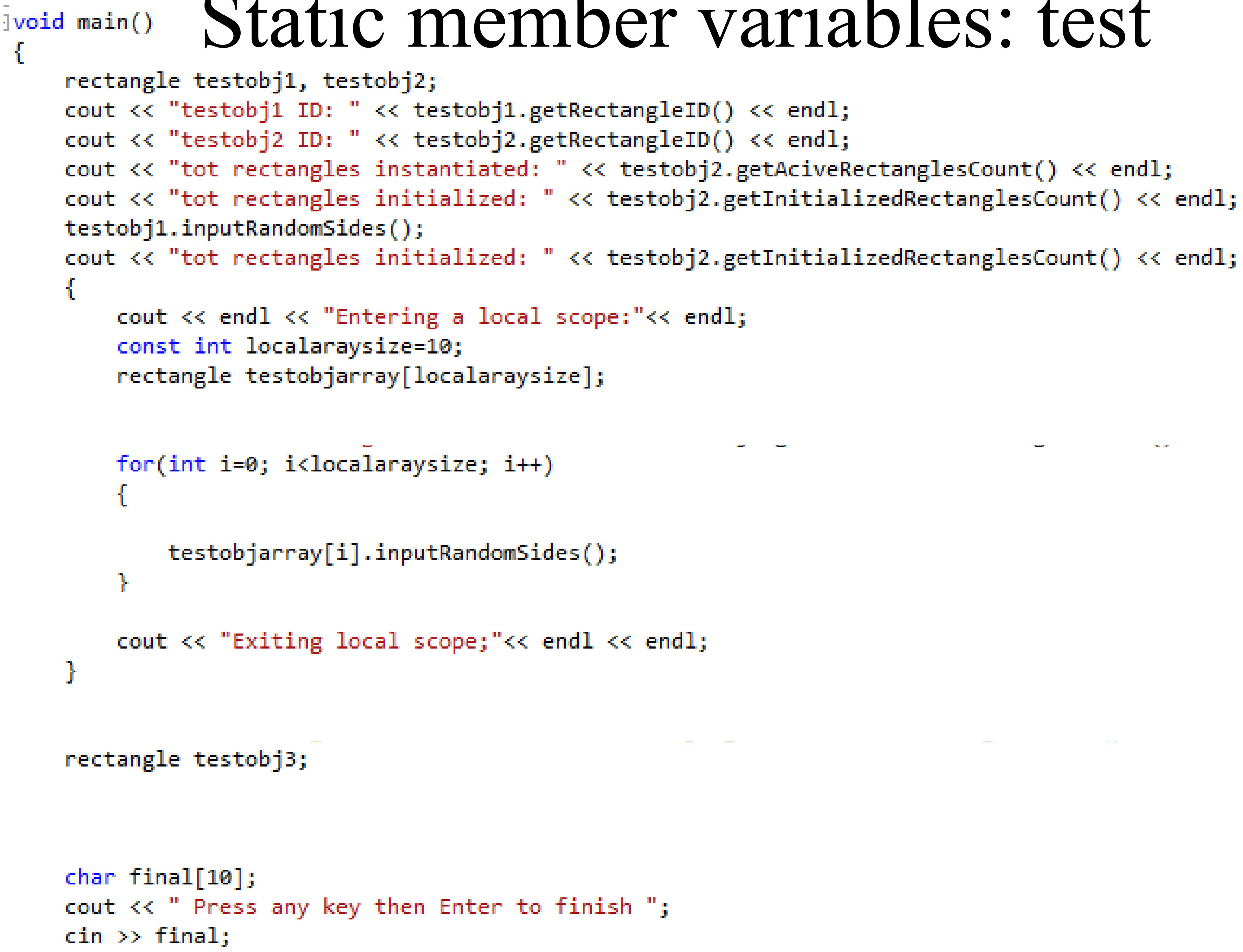
```
// copy constructor
rectangle(const rectangle& source) { ... }
// gets
double getArea() { ... }
double getPerimeter() { ... }
bool isInitialized(){return init_flag;}
double getSide(int sidenum) { ... }
void inputSides(double in_sideA, double in_sideB) { ... }
void inputSidesFromKeyboard() { ... }
void printRectangleInfo() { ... }
void inputRandomSides(double max_val=100) { ... }
void resetRectangle() { ... }
int getRectangleID(){return rectangle_ID;}
int getAciveRectanglesCount(){return rectangle_instances_alive_count;}
int getInitializedRectanglesCount(){return initialized rectangle instances count;}
```

Static member variables: test



```
void main()
{
    rectangle testobj1, testobj2;




    testobj1.inputRandomSides();

    {
        cout << endl << "Entering a local scope:"<< endl;
        const int localaraysize=10;
        rectangle testobjarray[localaraysize];


        for(int i=0; i<localaraysize; i++)
        {

            testobjarray[i].inputRandomSides();
        }

        cout << "Exiting local scope;"<< endl << endl;
    }


    rectangle testobj3;



    char final[10];
    cout << " Press any key then Enter to finish ";
    cin >> final;
```

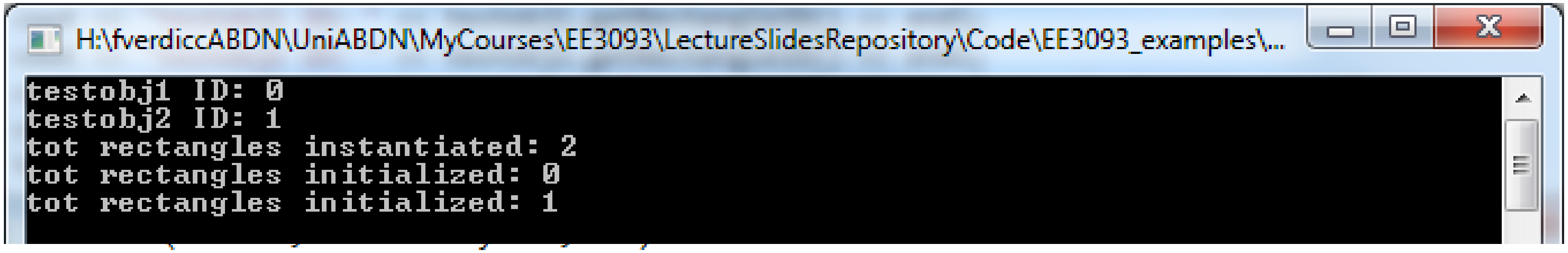
Static member variables: test



```
void main()
{
    rectangle testobj1, testobj2;
    cout << "testobj1 ID: " << testobj1.getRectangleID() << endl;
    cout << "testobj2 ID: " << testobj2.getRectangleID() << endl;
    cout << "tot rectangles instantiated: " << testobj2.getAciveRectanglesCount() << endl;
    cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
    testobj1.inputRandomSides();
    cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
    {
        cout << endl << "Entering a local scope:"<< endl;
        const int localaraysize=10;
        rectangle testobjarray[localaraysize];


        for(int i=0; i<localaraysize; i++)
        {

            testobjarray[i].inputRandomSides();
        }

        cout << "Exiting local scope;"<< endl << endl;
    }


    rectangle testobj3;



    char final[10];
    cout << " Press any key then Enter to finish ";
    cin >> final;
```

Static member variables: test

```
void main()
{
    rectangle testobj1, testobj2;
    cout << "testobj1 ID: " << testobj1.getRectangleID() << endl;
    cout << "testobj2 ID: " << testobj2.getRectangleID() << endl;
    cout << "tot rectangles instantiated: " << testobj2.getAciveRectanglesCount() << endl;
    cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
    testobj1.inputRandomSides();
    cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
```

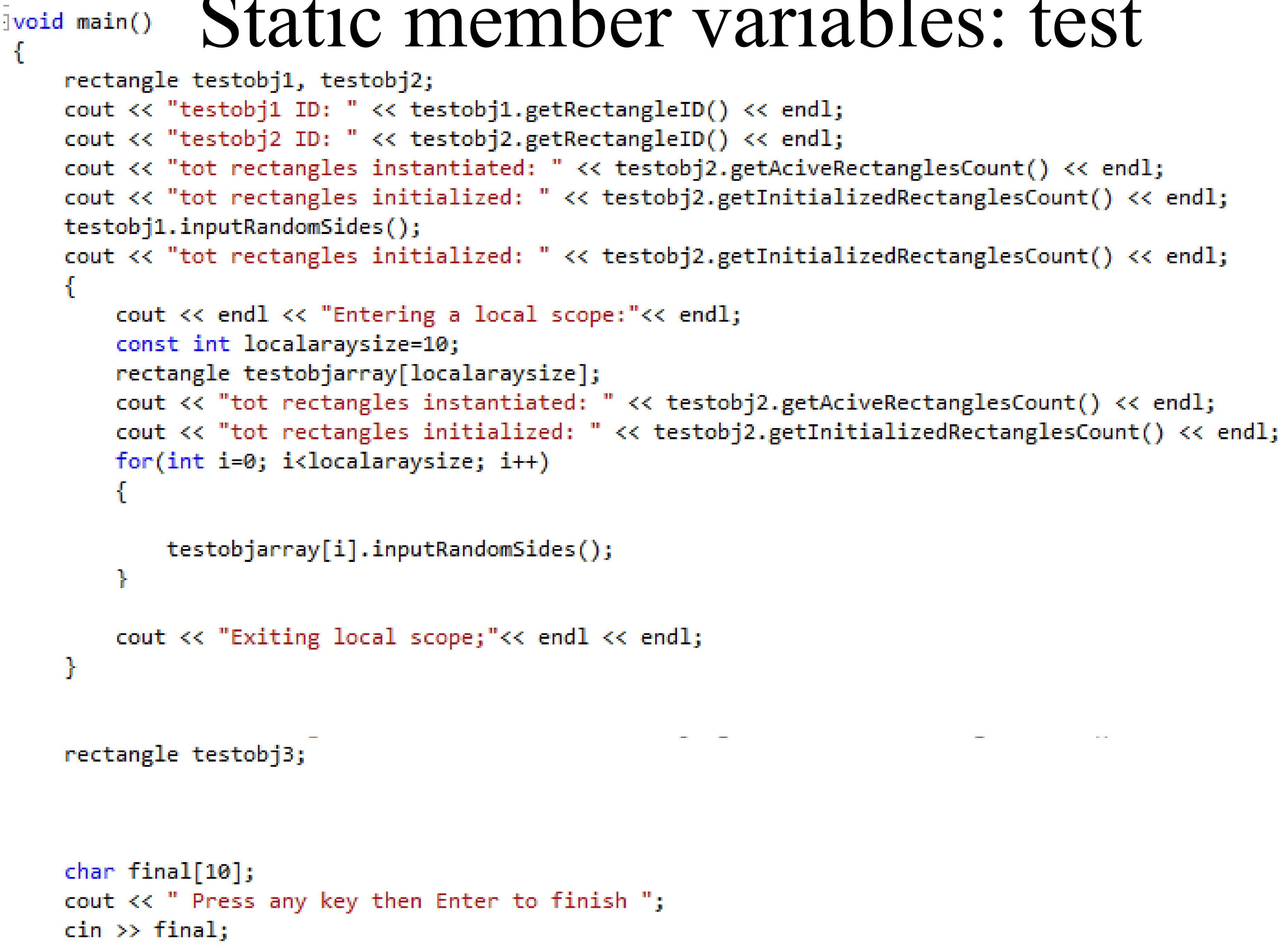


H:\fverdiccABDN\UniABDN\MyCourses\EE3093\LectureSlidesRepository\Code\EE3093_examples\...

```
testobj1 ID: 0
testobj2 ID: 1
tot rectangles instantiated: 2
tot rectangles initialized: 0
tot rectangles initialized: 1
```

```
        {

            testobjarray[i].inputRandomSides();
        }

        cout << "Exiting local scope;"<< endl << endl;
    }


    rectangle testobj3;



    char final[10];
    cout << " Press any key then Enter to finish ";
    cin >> final;
```

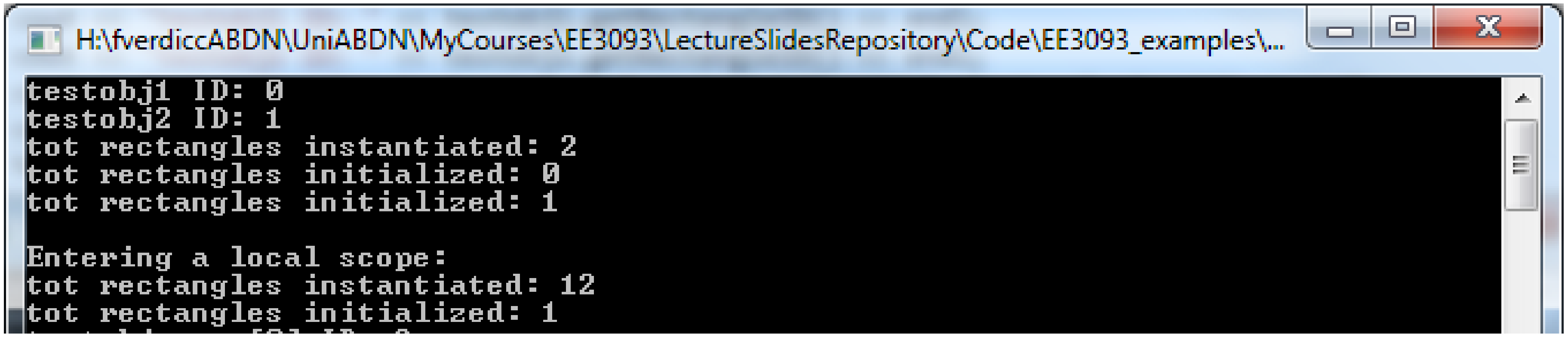
Static member variables: test



```cpp
void main()
{
    rectangle testobj1, testobj2;
    cout << "testobj1 ID: " << testobj1.getRectangleID() << endl;
    cout << "testobj2 ID: " << testobj2.getRectangleID() << endl;
    cout << "tot rectangles instantiated: " << testobj2.getAciveRectanglesCount() << endl;
    cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
    testobj1.inputRandomSides();
    cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
    {
        cout << endl << "Entering a local scope:"<< endl;
        const int localaraysize=10;
        rectangle testobjarray[localaraysize];
        cout << "tot rectangles instantiated: " << testobj2.getAciveRectanglesCount() << endl;
        cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
        for(int i=0; i<localaraysize; i++)
        {

            testobjarray[i].inputRandomSides();
        }

        cout << "Exiting local scope;"<< endl << endl;
    }


    rectangle testobj3;



    char final[10];
    cout << " Press any key then Enter to finish ";
    cin >> final;
```

Static member variables: test

```
void main()
{
    rectangle testobj1, testobj2;
    cout << "testobj1 ID: " << testobj1.getRectangleID() << endl;
    cout << "testobj2 ID: " << testobj2.getRectangleID() << endl;
    cout << "tot rectangles instantiated: " << testobj2.getAciveRectanglesCount() << endl;
    cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
    testobj1.inputRandomSides();
    cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
    {
        cout << endl << "Entering a local scope:"<< endl;
        const int localaraysize=10;
        rectangle testobjarray[localaraysize];
        cout << "tot rectangles instantiated: " << testobj2.getAciveRectanglesCount() << endl;
        cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
```

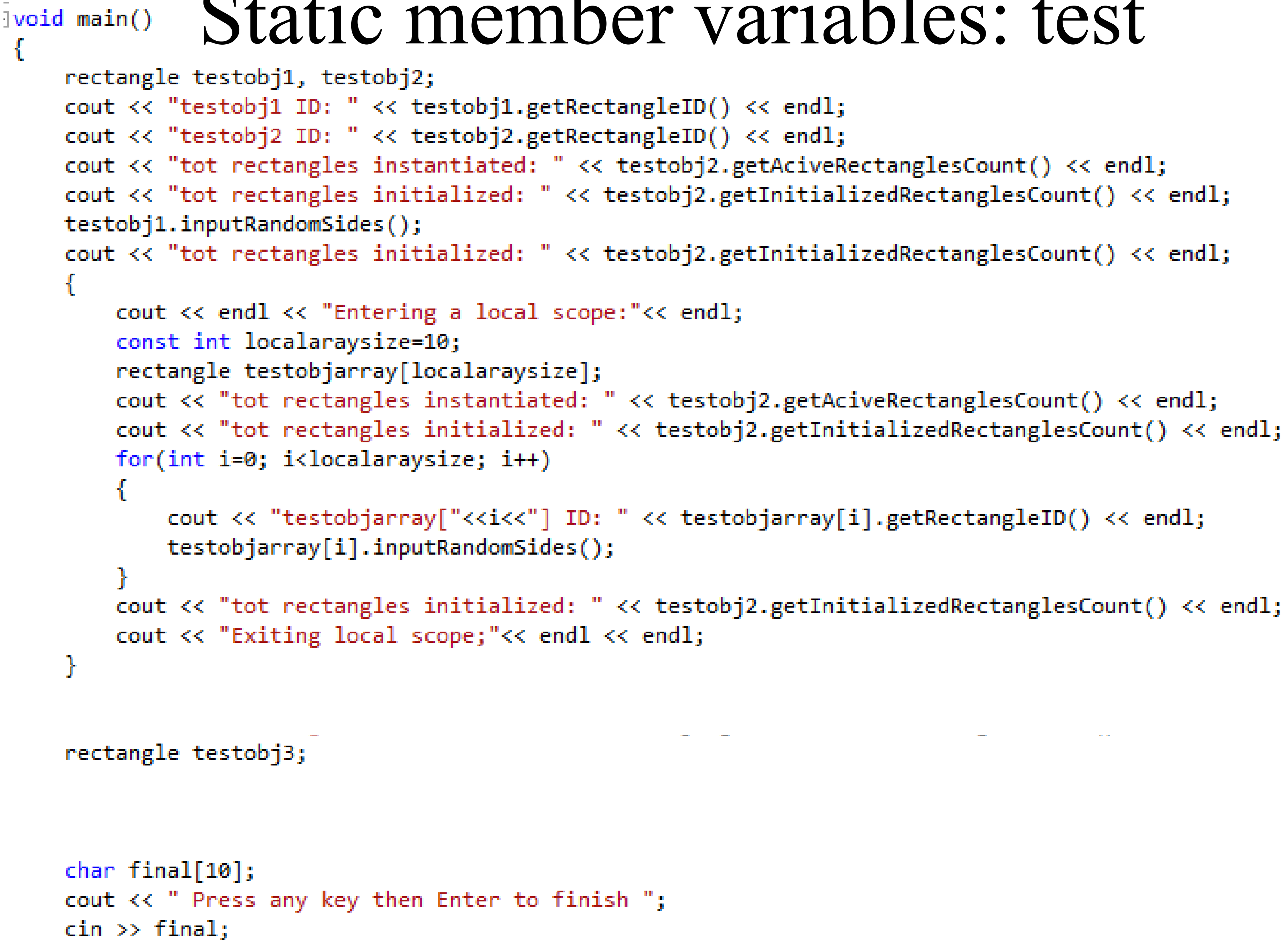


H:\fverdiccABDN\UniABDN\MyCourses\EE3093\LectureSlidesRepository\Code\EE3093_examples\...

```
testobj1 ID: 0
testobj2 ID: 1
tot rectangles instantiated: 2
tot rectangles initialized: 0
tot rectangles initialized: 1

Entering a local scope:
tot rectangles instantiated: 12
tot rectangles initialized: 1
```

```
    rectangle testobj3;



    char final[10];
    cout << " Press any key then Enter to finish ";
    cin >> final;
```

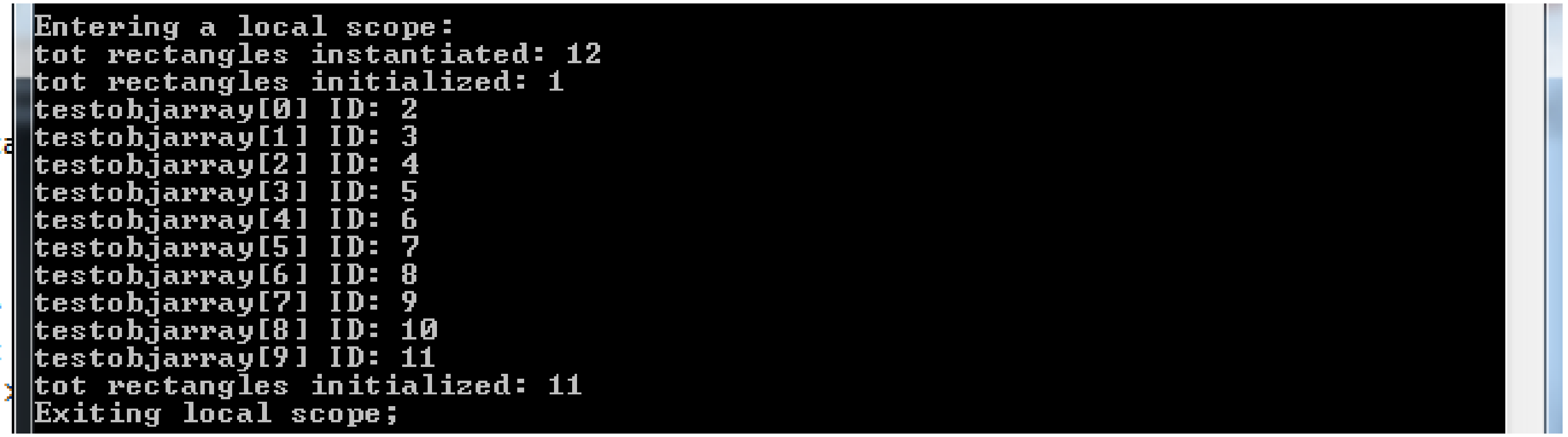
Static member variables: test



```cpp
void main()
{
    rectangle testobj1, testobj2;
    cout << "testobj1 ID: " << testobj1.getRectangleID() << endl;
    cout << "testobj2 ID: " << testobj2.getRectangleID() << endl;
    cout << "tot rectangles instantiated: " << testobj2.getAciveRectanglesCount() << endl;
    cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
    testobj1.inputRandomSides();
    cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
    {
        cout << endl << "Entering a local scope:"<< endl;
        const int localaraysize=10;
        rectangle testobjarray[localaraysize];
        cout << "tot rectangles instantiated: " << testobj2.getAciveRectanglesCount() << endl;
        cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
        for(int i=0; i<localaraysize; i++)
        {
            cout << "testobjarray["<<i<<"] ID: " << testobjarray[i].getRectangleID() << endl;
            testobjarray[i].inputRandomSides();
        }
        cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
        cout << "Exiting local scope;"<< endl << endl;
    }


    rectangle testobj3;



    char final[10];
    cout << " Press any key then Enter to finish ";
    cin >> final;
```

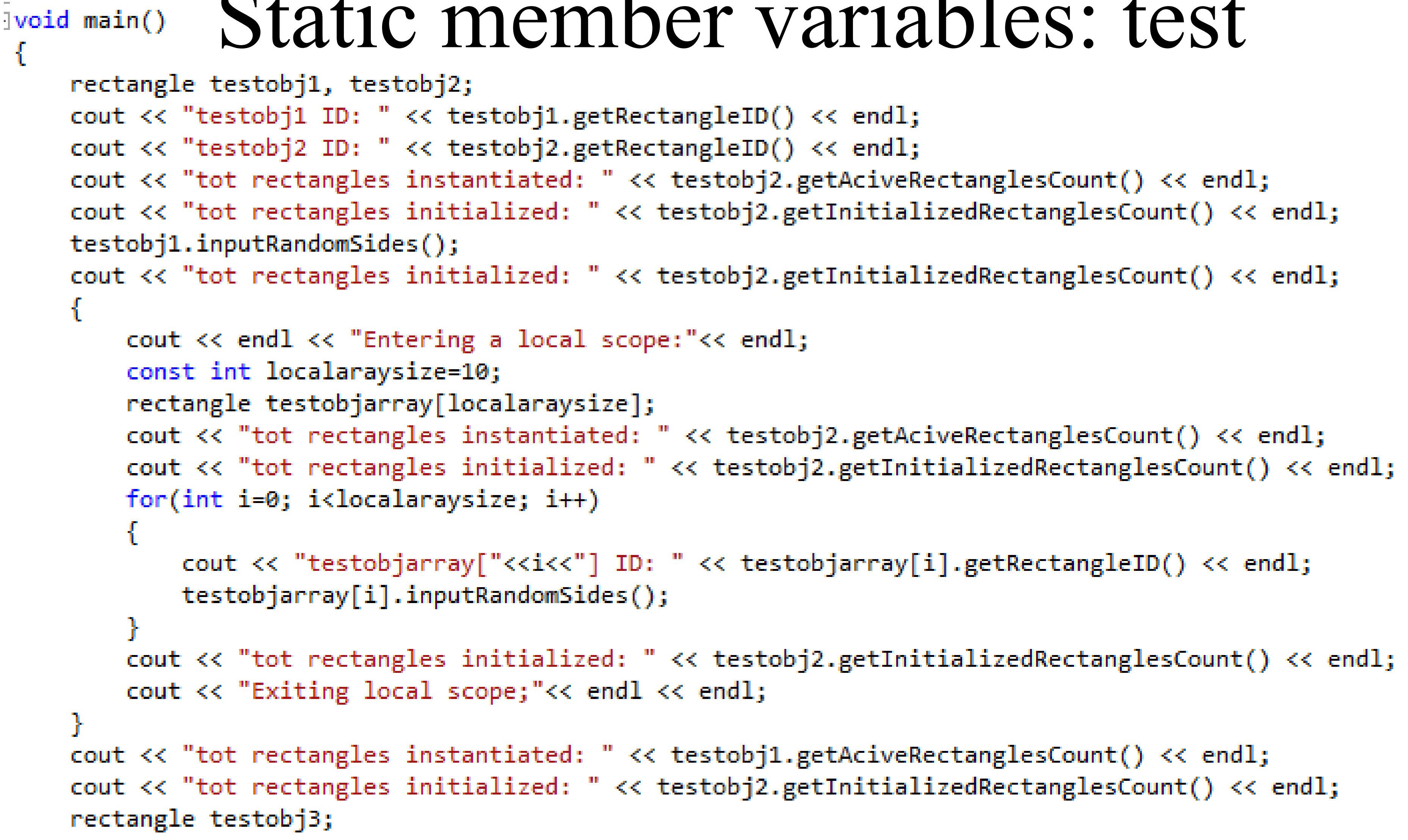
Static member variables: test

```
void main()
{
    rectangle testobj1, testobj2;
    cout << "testobj1 ID: " << testobj1.getRectangleID() << endl;
    cout << "testobj2 ID: " << testobj2.getRectangleID() << endl;
    cout << "tot rectangles instantiated: " << testobj2.getAciveRectanglesCount() << endl;
    cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
    testobj1.inputRandomSides();
    cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
    {
        cout << endl << "Entering a local scope:"<< endl;
        const int localaraysize=10;
        rectangle testobjarray[localaraysize];
        cout << "tot rectangles instantiated: " << testobj2.getAciveRectanglesCount() << endl;
        cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
        for(int i=0; i<localaraysize; i++)
        {
            cout << "testobjarray["<<i<<"] ID: " << testobjarray[i].getRectangleID() << endl;
            testobjarray[i].inputRandomSides();
        }
        cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
        cout << "Exiting local scope;"<< endl << endl;
    }

    recta


    char
    cout
    cin >
```

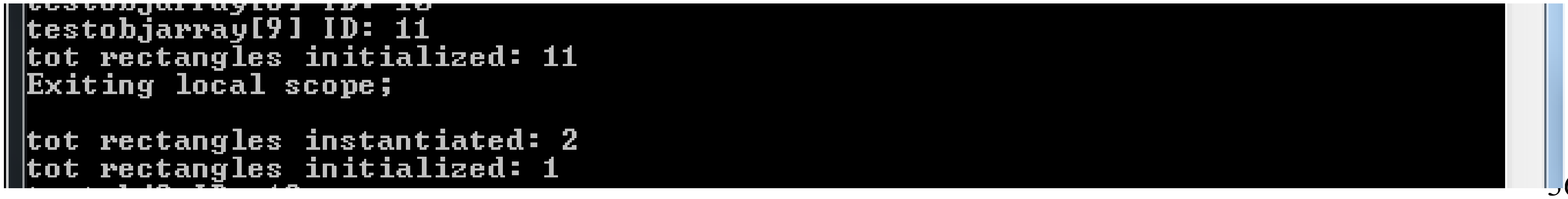


```
Entering a local scope:
tot rectangles instantiated: 12
tot rectangles initialized: 1
testobjarray[0] ID: 2
testobjarray[1] ID: 3
testobjarray[2] ID: 4
testobjarray[3] ID: 5
testobjarray[4] ID: 6
testobjarray[5] ID: 7
testobjarray[6] ID: 8
testobjarray[7] ID: 9
testobjarray[8] ID: 10
testobjarray[9] ID: 11
tot rectangles initialized: 11
Exiting local scope;
```

Static member variables: test



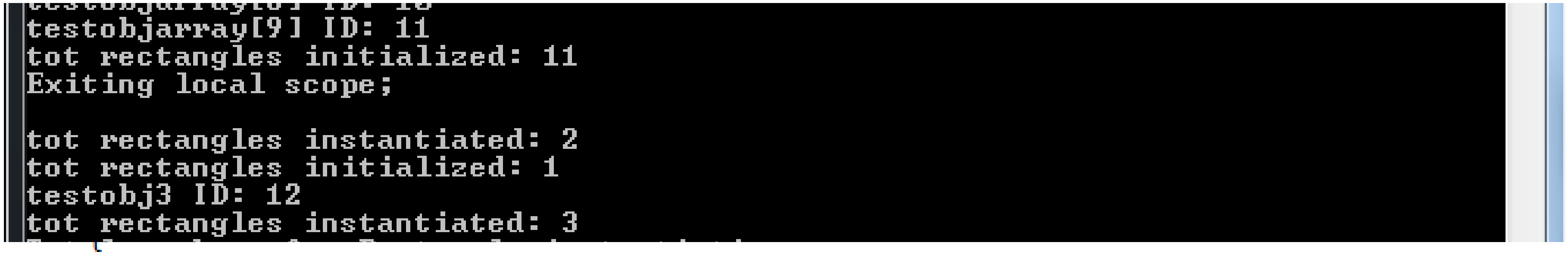
```
void main()
{
    rectangle testobj1, testobj2;
    cout << "testobj1 ID: " << testobj1.getRectangleID() << endl;
    cout << "testobj2 ID: " << testobj2.getRectangleID() << endl;
    cout << "tot rectangles instantiated: " << testobj2.getAciveRectanglesCount() << endl;
    cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
    testobj1.inputRandomSides();
    cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
    {
        cout << endl << "Entering a local scope:"<< endl;
        const int localaraysize=10;
        rectangle testobjarray[localaraysize];
        cout << "tot rectangles instantiated: " << testobj2.getAciveRectanglesCount() << endl;
        cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
        for(int i=0; i<localaraysize; i++)
        {
            cout << "testobjarray["<<i<<"] ID: " << testobjarray[i].getRectangleID() << endl;
            testobjarray[i].inputRandomSides();
        }
        cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
        cout << "Exiting local scope;"<< endl << endl;
    }
    cout << "tot rectangles instantiated: " << testobj1.getAciveRectanglesCount() << endl;
    cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
    rectangle testobj3;
```



```
testobjarray[9] ID: 11
tot rectangles initialized: 11
Exiting local scope;

tot rectangles instantiated: 2
tot rectangles initialized: 1
```

Static member variables: test

```
void main()
{
    rectangle testobj1, testobj2;
    cout << "testobj1 ID: " << testobj1.getRectangleID() << endl;
    cout << "testobj2 ID: " << testobj2.getRectangleID() << endl;
    cout << "tot rectangles instantiated: " << testobj2.getAciveRectanglesCount() << endl;
    cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
    testobj1.inputRandomSides();
    cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
    {
```

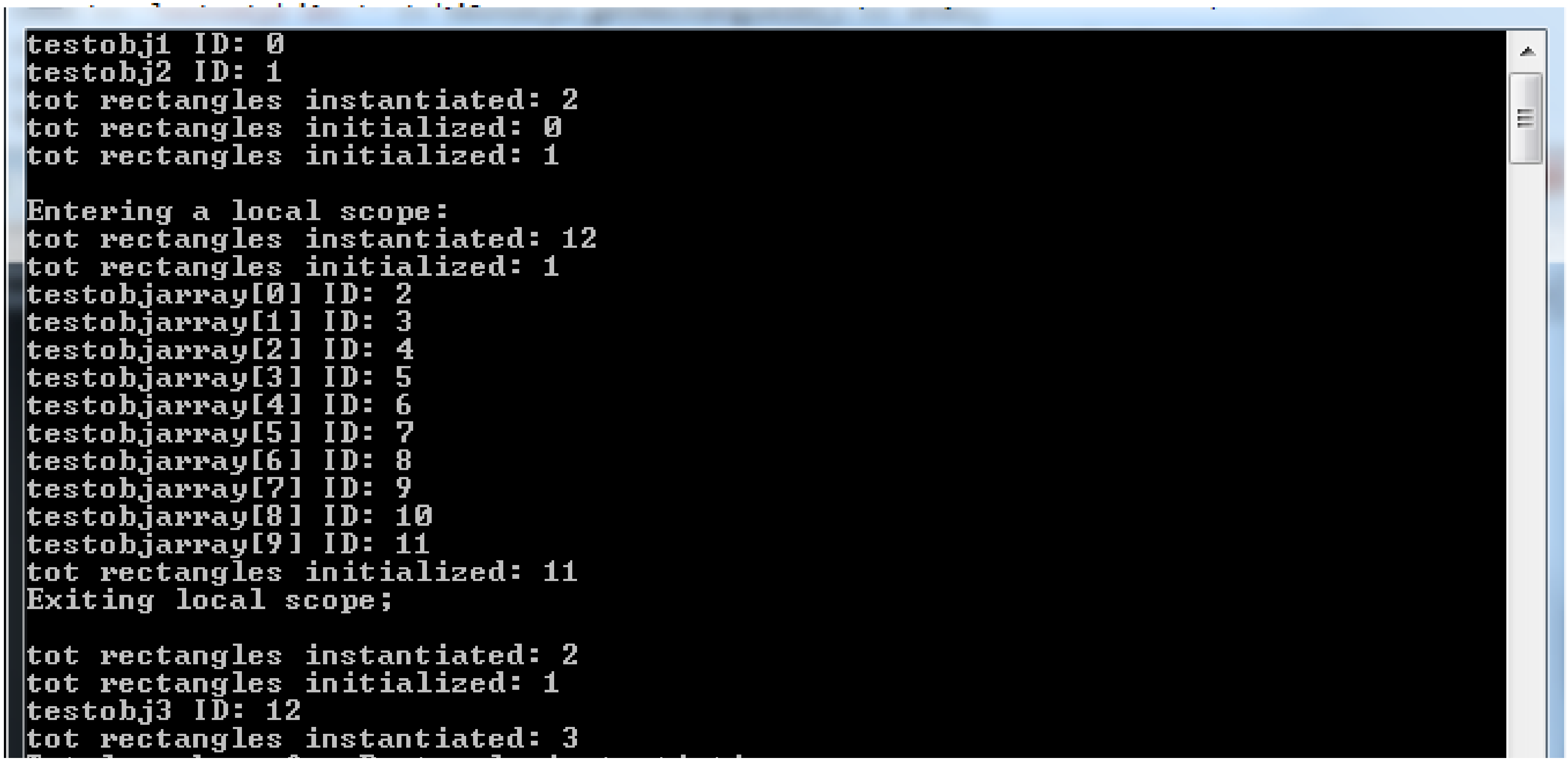


```
testobjarray[9] ID: 11
tot rectangles initialized: 11
Exiting local scope;

tot rectangles instantiated: 2
tot rectangles initialized: 1
testobj3 ID: 12
tot rectangles instantiated: 3
```

```
            cout << "testobjarray["<<i<<"] ID: " << testobjarray[i].getRectangleID() << endl;
            testobjarray[i].inputRandomSides();
        }
        cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
        cout << "Exiting local scope;"<< endl << endl;
    }
    cout << "tot rectangles instantiated: " << testobj1.getAciveRectanglesCount() << endl;
    cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
    rectangle testobj3;
    cout << "testobj3 ID: " << testobj3.getRectangleID() << endl;
    cout << "tot rectangles instantiated: " << testobj1.getAciveRectanglesCount() << endl;

    char final[10];
    cout << " Press any key then Enter to finish ";
    cin >> final;
```

Static member variables: test

```
void main()
{
```



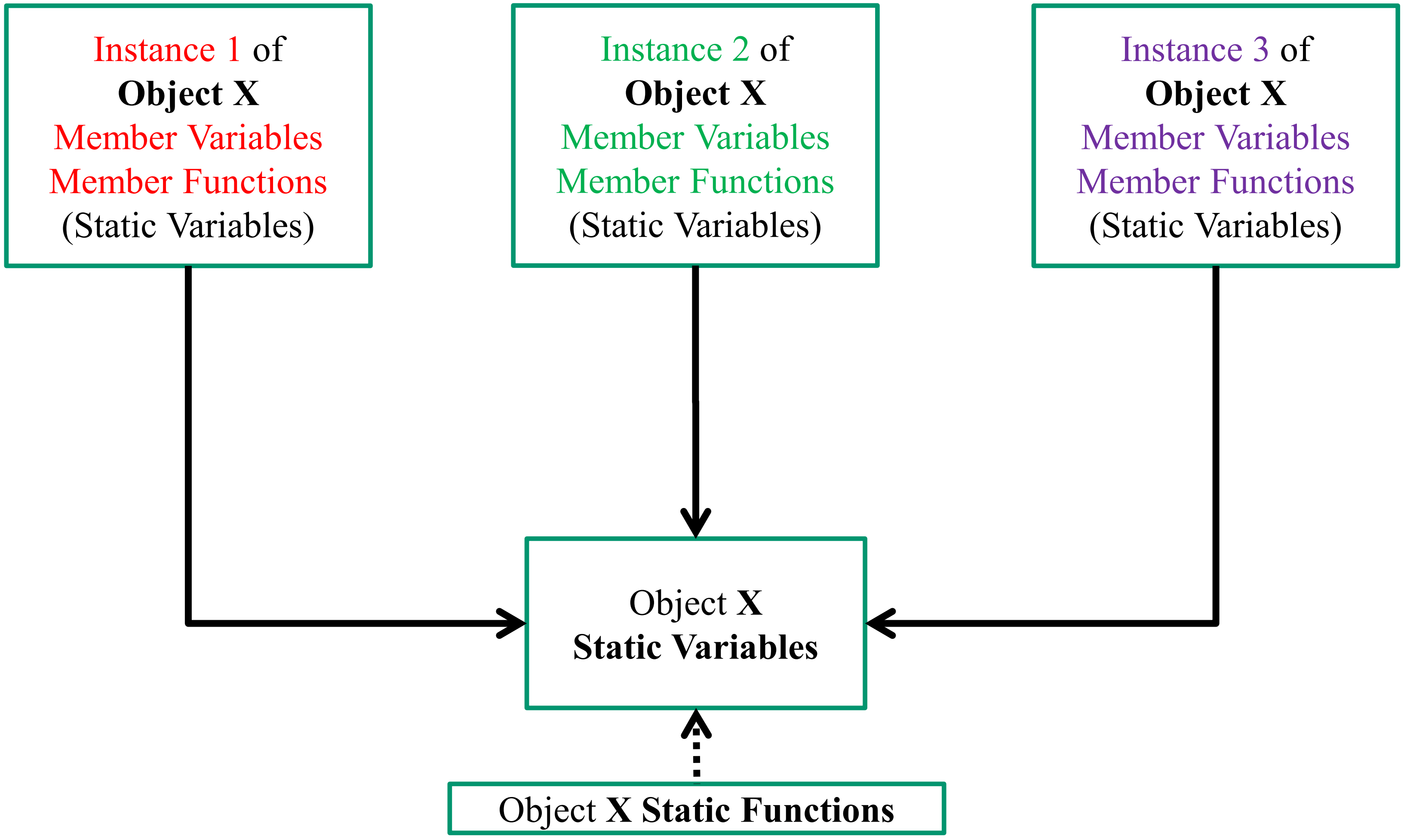
```
testobj1 ID: 0
testobj2 ID: 1
tot rectangles instantiated: 2
tot rectangles initialized: 0
tot rectangles initialized: 1

Entering a local scope:
tot rectangles instantiated: 12
tot rectangles initialized: 1
testobjarray[0] ID: 2
testobjarray[1] ID: 3
testobjarray[2] ID: 4
testobjarray[3] ID: 5
testobjarray[4] ID: 6
testobjarray[5] ID: 7
testobjarray[6] ID: 8
testobjarray[7] ID: 9
testobjarray[8] ID: 10
testobjarray[9] ID: 11
tot rectangles initialized: 11
Exiting local scope;

tot rectangles instantiated: 2
tot rectangles initialized: 1
testobj3 ID: 12
tot rectangles instantiated: 3
```

```
    cout << "tot rectangles instantiated: " << testobj1.getAciveRectanglesCount() << endl;
    cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
    rectangle testobj3;
    cout << "testobj3 ID: " << testobj3.getRectangleID() << endl;
    cout << "tot rectangles instantiated: " << testobj1.getAciveRectanglesCount() << endl;

    char final[10];
    cout << " Press any key then Enter to finish ";
    cin >> final;
```

Any question?

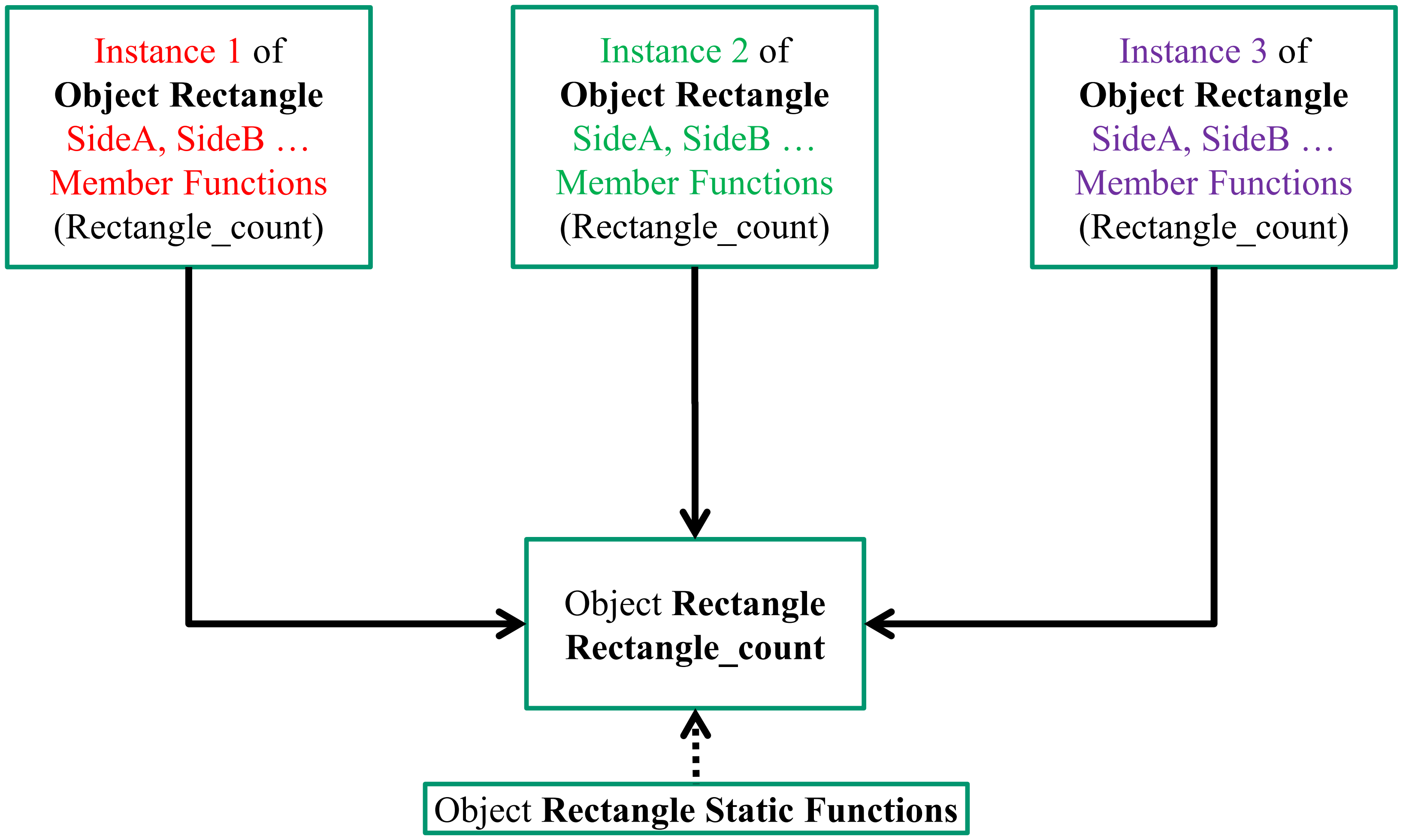
Static member functions

A **static member function** is independent of any instances of the class (can be used without creating an instance of that class) and can only use static variables of that class:

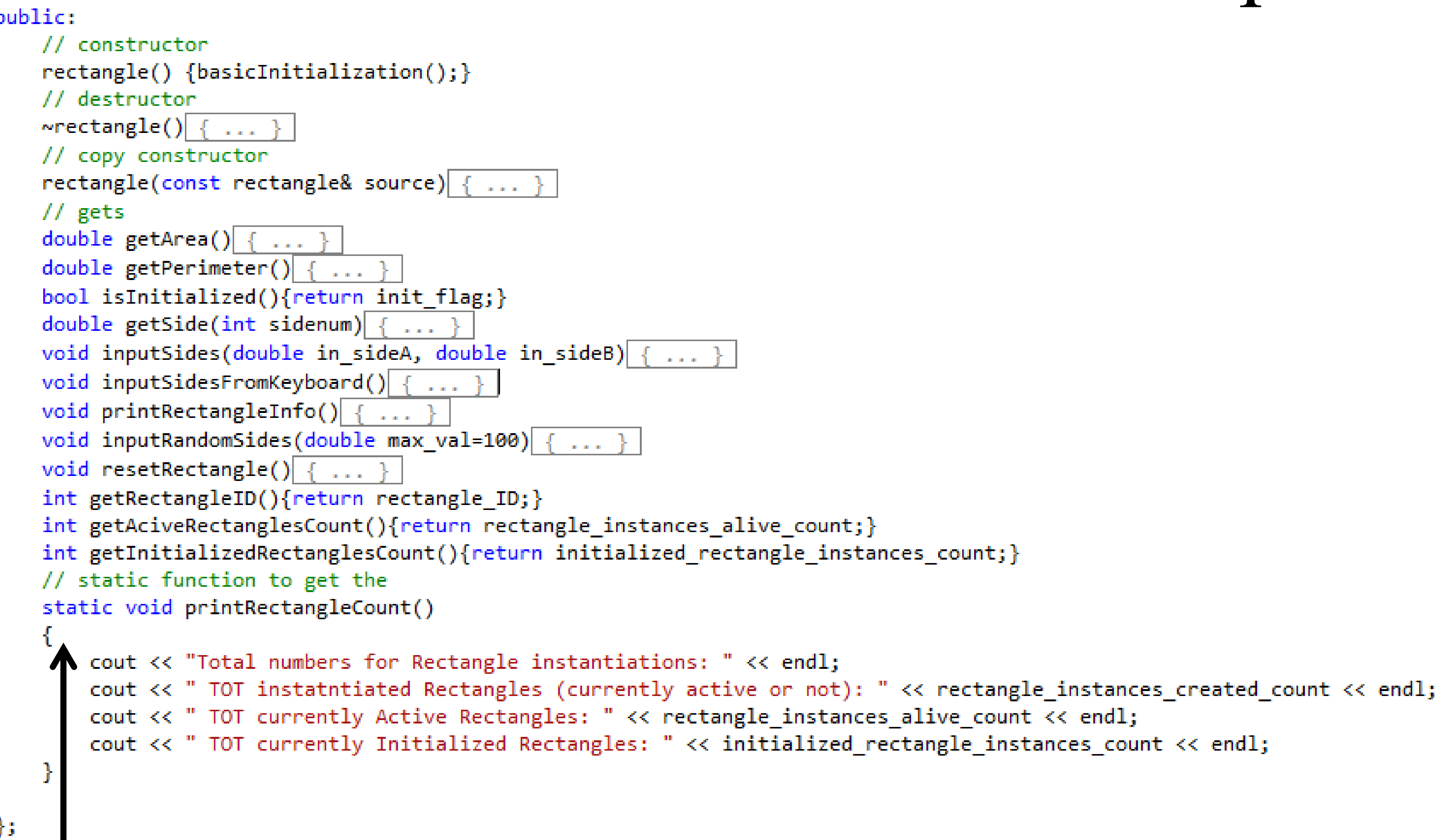


Static member functions

A **static member function** is independent of any instances of the class (can be used without creating an instance of that class) and can only use static variables of that class:



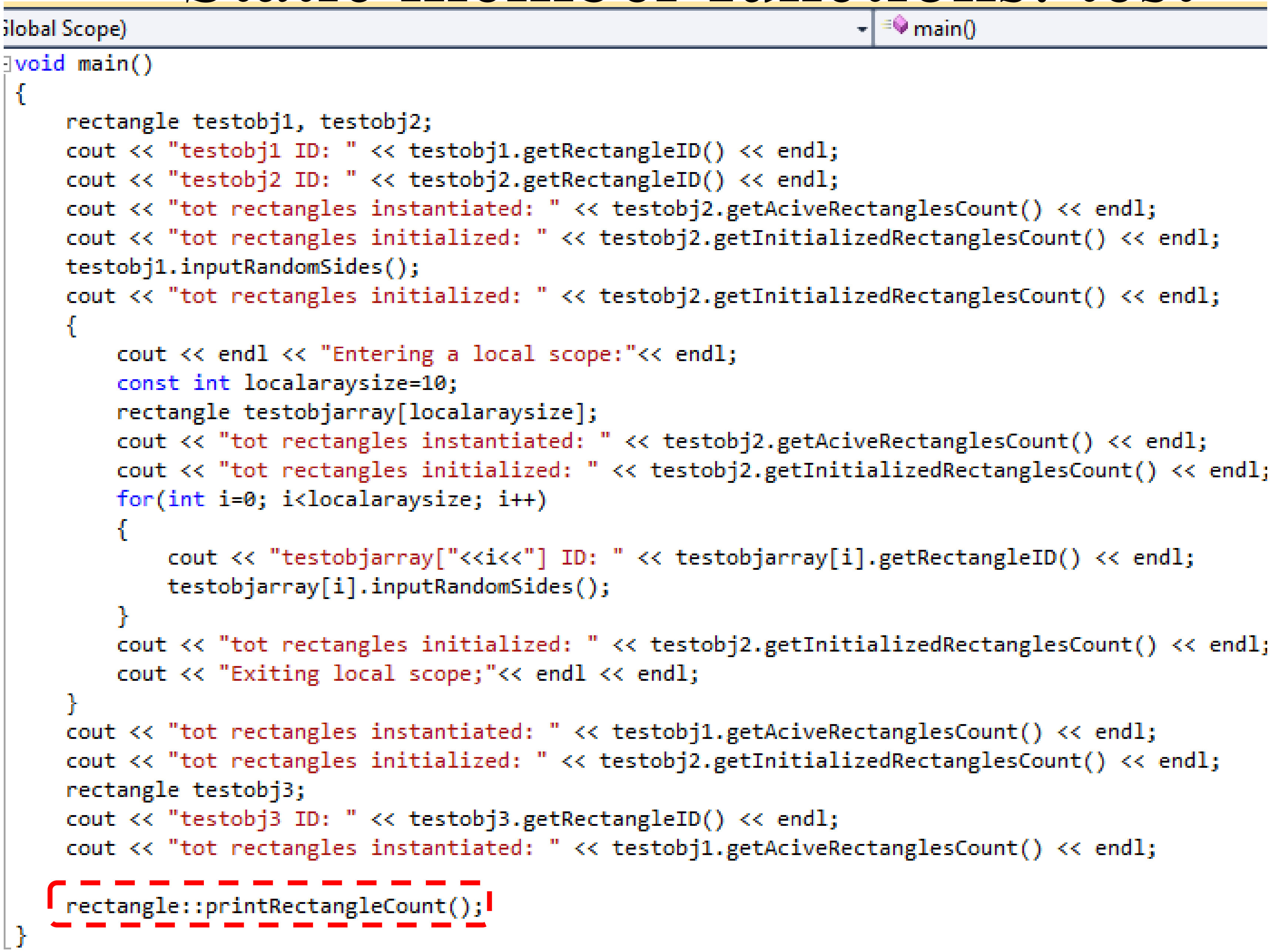
Static member functions: example



```
public:
    // constructor
    rectangle() {basicInitialization();}
    // destructor
    ~rectangle() { ... }
    // copy constructor
    rectangle(const rectangle& source) { ... }
    // gets
    double getArea() { ... }
    double getPerimeter() { ... }
    bool isInitialized(){return init_flag;}
    double getSide(int sidenum) { ... }
    void inputSides(double in_sideA, double in_sideB) { ... }
    void inputSidesFromKeyboard() { ... }
    void printRectangleInfo() { ... }
    void inputRandomSides(double max_val=100) { ... }
    void resetRectangle() { ... }
    int getRectangleID(){return rectangle_ID;}
    int getAciveRectanglesCount(){return rectangle_instances_alive_count;}
    int getInitializedRectanglesCount(){return initialized_rectangle_instances_count;}
    // static function to get the
    static void printRectangleCount()
    {
        cout << "Total numbers for Rectangle instantiations: " << endl;
        cout << " TOT instatntiated Rectangles (currently active or not): " << rectangle_instances_created_count << endl;
        cout << " TOT currently Active Rectangles: " << rectangle_instances_alive_count << endl;
        cout << " TOT currently Initialized Rectangles: " << initialized_rectangle_instances_count << endl;
    }

};
```

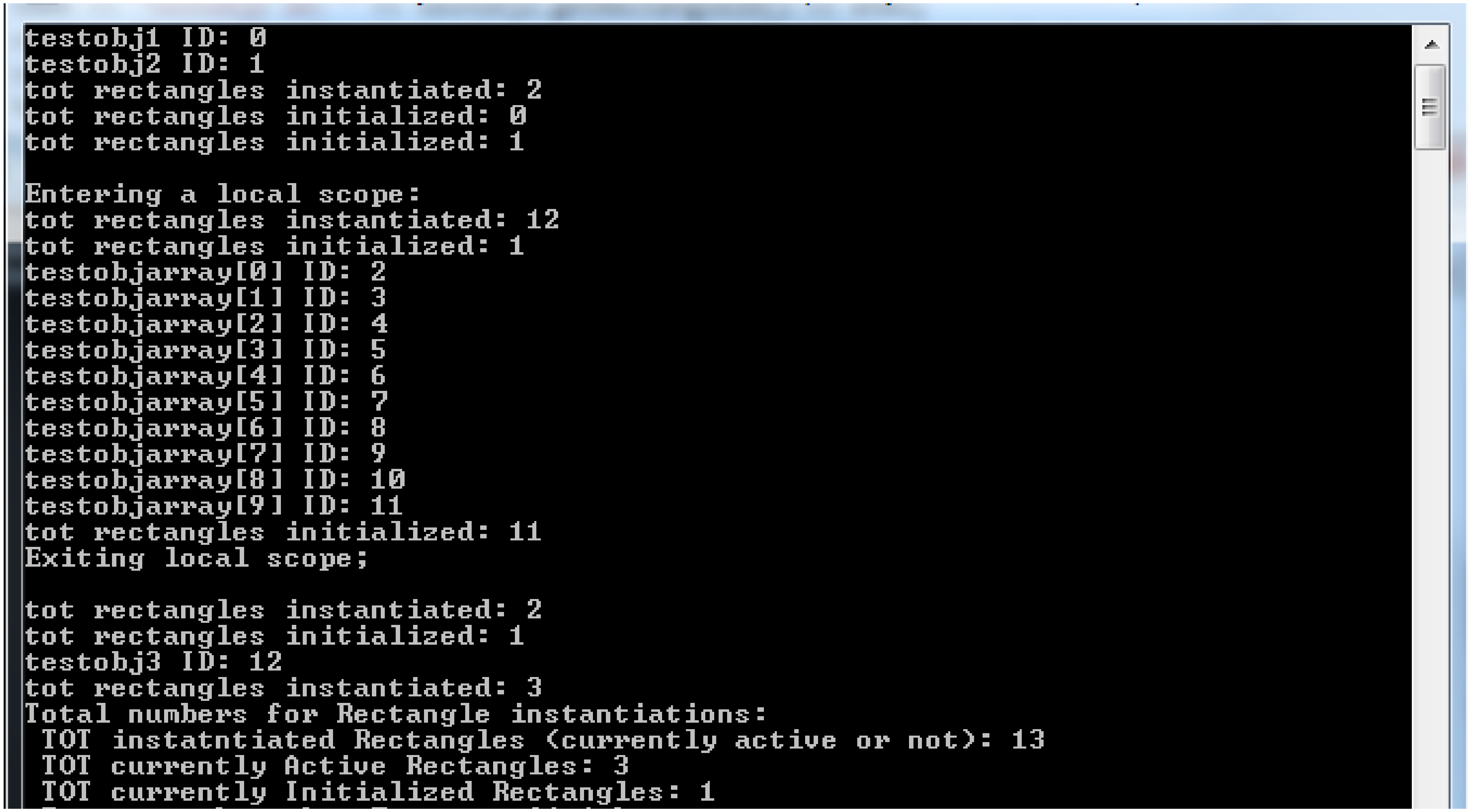
Keyword **static**; can only use static member variables (and functions)

Static member functions: test



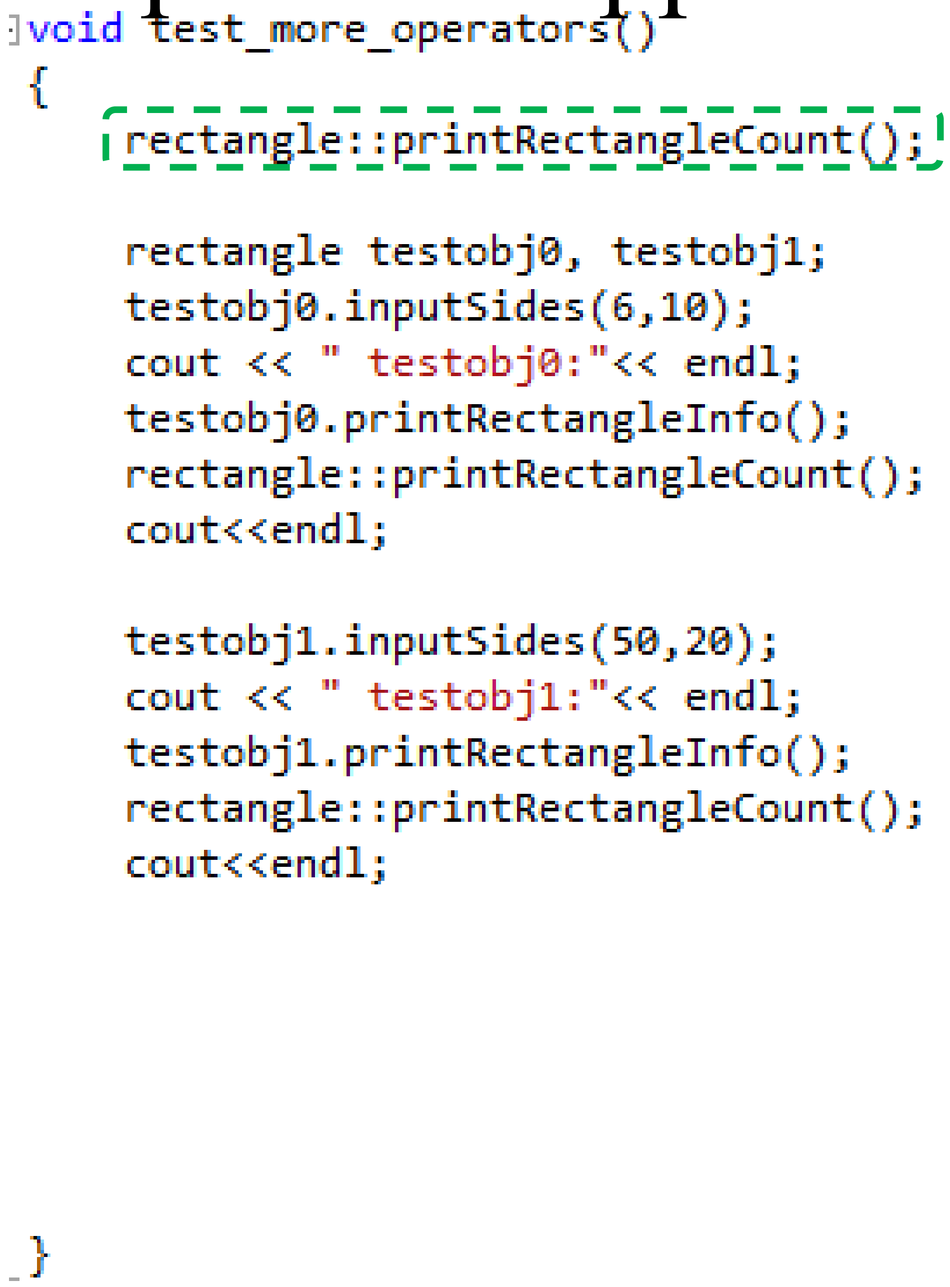
```
void main()
{
    rectangle testobj1, testobj2;
    cout << "testobj1 ID: " << testobj1.getRectangleID() << endl;
    cout << "testobj2 ID: " << testobj2.getRectangleID() << endl;
    cout << "tot rectangles instantiated: " << testobj2.getAciveRectanglesCount() << endl;
    cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
    testobj1.inputRandomSides();
    cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
    {
        cout << endl << "Entering a local scope:"<< endl;
        const int localaraysize=10;
        rectangle testobjarray[localaraysize];
        cout << "tot rectangles instantiated: " << testobj2.getAciveRectanglesCount() << endl;
        cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
        for(int i=0; i<localaraysize; i++)
        {
            cout << "testobjarray["<<i<<"] ID: " << testobjarray[i].getRectangleID() << endl;
            testobjarray[i].inputRandomSides();
        }
        cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
        cout << "Exiting local scope;"<< endl << endl;
    }
    cout << "tot rectangles instantiated: " << testobj1.getAciveRectanglesCount() << endl;
    cout << "tot rectangles initialized: " << testobj2.getInitializedRectanglesCount() << endl;
    rectangle testobj3;
    cout << "testobj3 ID: " << testobj3.getRectangleID() << endl;
    cout << "tot rectangles instantiated: " << testobj1.getAciveRectanglesCount() << endl;

    rectangle::printRectangleCount();
}
```

Static member functions: test



```
testobj1 ID: 0
testobj2 ID: 1
tot rectangles instantiated: 2
tot rectangles initialized: 0
tot rectangles initialized: 1

Entering a local scope:
tot rectangles instantiated: 12
tot rectangles initialized: 1
testobjarray[0] ID: 2
testobjarray[1] ID: 3
testobjarray[2] ID: 4
testobjarray[3] ID: 5
testobjarray[4] ID: 6
testobjarray[5] ID: 7
testobjarray[6] ID: 8
testobjarray[7] ID: 9
testobjarray[8] ID: 10
testobjarray[9] ID: 11
tot rectangles initialized: 11
Exiting local scope;

tot rectangles instantiated: 2
tot rectangles initialized: 1
testobj3 ID: 12
tot rectangles instantiated: 3
Total numbers for Rectangle instantiations:
 TOT instatntiated Rectangles (currently active or not): 13
 TOT currently Active Rectangles: 3
 TOT currently Initialized Rectangles: 1
```

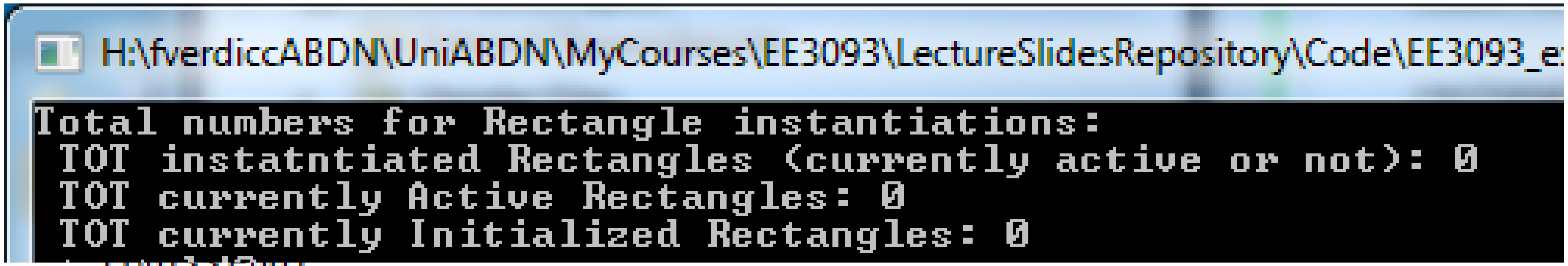
Operators applied to objects: example



```
void test_more_operators()
{
    rectangle::printRectangleCount();

    rectangle testobj0, testobj1;
    testobj0.inputSides(6,10);
    cout << " testobj0:"<< endl;
    testobj0.printRectangleInfo();
    rectangle::printRectangleCount();
    cout<<endl;

    testobj1.inputSides(50,20);
    cout << " testobj1:"<< endl;
    testobj1.printRectangleInfo();
    rectangle::printRectangleCount();
    cout<<endl;




}
```

Operators applied to objects: example

```
void test_more_operators()
{
    rectangle::printRectangleCount();
```



H:\fverdiccABDN\UniABDN\MyCourses\EE3093\LectureSlidesRepository\Code\EE3093_e

```
Total numbers for Rectangle instantiations:
 TOT instatntiated Rectangles (currently active or not): 0
 TOT currently Active Rectangles: 0
 TOT currently Initialized Rectangles: 0
```

```
    cout<<endl;

    testobj1.inputSides(50,20);
    cout << " testobj1:"<< endl;
    testobj1.printRectangleInfo();
    rectangle::printRectangleCount();
    cout<<endl;


}
```

Any question?