

University American College Skopje

Course: Object Programming

Constructors and Destructors

Exercises

Prepared by: Biljana Stojčevska, Ph.D.,
Adrijan Božinovski, Ph.D.

Assignment 1

- Write a program with class **my_class** with an integer data member **a**, a default constructor, a parameterized constructor, a destructor and a function member that displays the value of **a**.

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

```
//definiton of my_class
class my_class {
    int a; //data member a
public:
    my_class(int x); //constructor
    declaration
    ~my_class(); //desctructor
    declaration

    void show(); //declaration of
    show
};
```

```
//default constructor definition
my_class::my_class(int x){
    cout<<"In the default
    constructor\n";
    cout<<"x=10"<<endl;
    a=10;
}
```

```
//parameterized constructor
definition
my_class::my_class(int x){
    cout<<"In the
    parameterized constructor\n";
    cout<<"x="<<x<<endl;
    a=x;
}
```

```
//destructor definition
my_class::~~my_class(){
    cout<<"Destroying\n";
}

//definition of show
void my_class::show(){
    cout << a <<endl;
}
```

```
void main(){
    my_class ob1;
    ob1.show();
    cout << endl;
    my_class ob2(15);
    ob2.show();
}
```

Assignment 1

- Output

```
In the default constructor
```

```
x = 10
```

```
10
```

```
In the parameterized constructor
```

```
x = 15
```

```
15
```

```
Destroying
```

```
Destroying
```

```
Press any key to continue
```

Assignment 2

- Create a class called **CreateAndDestroy** that will have a single integer data field.
- Its constructor will be parameterized and will set the data field, and also display a message that the object with the given value for the data field is created.
- Its destructor will display a message that the object with the given data field is destroyed

Assignment 2

- Create a global void function **f()** in which you will create
 - A CreateAndDestroy object
 - A static CreateAndDestroy object
 - Another CreateAndDestroy object
- In the main() function
 1. Create a local CreateAndDestroy object
 2. Create a local static CreateAndDestroy object
 3. Invoke the f() function
 4. Create another local CreateAndDestroy object

Assignment 2

- Output

```
Object 1 is created    (global before main)
Object 2 is created    (local in main)
Object 3 is created    (local static in main)
Object 5 is created    (local in f() )
Object 6 is created    (local static in f() )
Object 7 is created    (local in f() )
Object 7 is destroyed
Object 5 is destroyed
Object 4 is created    (local in main)
Object 4 is destroyed
Object 2 is destroyed
Object 6 is destroyed
Object 3 is destroyed
Press any key to continue
```

Why is Object 1 not (said to be) destroyed?

Assignment 3

- Create a class **Car** that has two private data fields: integer liters and double consumption (liters/km)
- Its default constructor should set those values to 40 and 0.1, respectively. It should also have a parameterized constructor to set those values. Both constructors should display messages that a car with the given features had been created
- Its destructor should display a message that a car with the given features had been destroyed
- It should have a double maxDistance() method that will return the maximum

Assignment 3

- In the main() function
 - Create a Car using the default constructor and display the maximum distance it can travel
 - Create a Car using the parameterized constructor, using values 50 and 0.2 as parameters, and display the maximum distance it can travel
 - Create a pointer to a new Car created using the parameterized constructor, using values 60 and 0.25 as parameters, and display the maximum distance it can travel

Assignment 3

- Output

```
A car that has a fuel capacity of 40 liters and consumes 0.1 l/km is created
This car can drive at most 400 kilometers on a single tank
A car that has a fuel capacity of 50 liters and consumes 0.2 l/km is created
This car can drive at most 250 kilometers on a single tank
A car that has a fuel capacity of 60 liters and consumes 0.25 l/km is created
This car can drive at most 240 kilometers on a single tank
A car that has a fuel capacity of 50 liters and consumes 0.2 l/km is destroyed
A car that has a fuel capacity of 40 liters and consumes 0.1 l/km is destroyed
Press any key to continue.
```

Why is this car not destroyed????

Assignment 4

- Create a class **Student** which will have two private data fields: double gpa, and integer credits
- It should have functions getGPA() and getGrades(), which will return the respective values
- It should have a default constructor, which will set both values to zero, and a parameterized constructor, which will set them to the values of the constructor's

Assignment 4

- In the main() function, create a dynamically allocated array of Student objects (i.e. the user should input how many students there will be and there should be no limit on the possible array size)
 - **Student *s = new Student[number_of_students]**
is equivalent to:
Student *s = new Student()
[number_of_students];
 - This is why it's necessary to have a default constructor defined –so that the adequate room in memory would be reserved

Assignment 4

- For each student enter the GPA and the number of credits through the keyboard
 - Two ways (assuming the array of students is called **s**):

- **`s[i] = *(new Student(UserInputGPA, UserInputGrade));`**

or

- **`Student *sp = new Student(UserInputGPA, UserInputGrade);`**

`s[i] = *sp;`

`delete sp;`

`sp = NULL;`

Preferred way,
because of memory
clean-up possibility

- Then calculate the average GPA and the total

Assignment 4

- Output

```
How many students will there be? 2
Student 0 - GPA:      1.2
Student 0 - Credits:  3

Student 1 - GPA:      3.4
Student 1 - Credits:  5

The following students have been input:
Student 0      GPA: 1.2      Credits: 3
Student 1      GPA: 3.4      Credits: 5
The average GPA of all students is 2.3 and a total of 8 credits were achieved
Press any key to continue_
```

Assignment 5

- Modify assignment 4, so that messages would be displayed upon constructing elements with the default and the parameterized constructors
- Add a destructor, which should also display a message that a Student object is destroyed
- Also, messages should be displayed upon assigning values to the pointer to the new Student object, as well as to the element of the array of Student objects. Also, display a message upon the delete *arr;

Using Random Numbers

- NOTE – to use random numbers in C++:
 - **#include <ctime>** among the libraries that will be included in your code
 - At the beginning of the `main()` function, place the following statement:
srand(time(NULL));
This will set the randomizing seed to the current system time, producing truly random numbers
 - Then, the function **rand()** will give you a random integer from 0 to 32767 (i.e. the **RAND_MAX** constant). Using the modulus (%) operator, you may limit the range of the maximum random number obtained

Assignment 6

- Create a class **Polygon**. It should have the following members:
 - Public:
 - A double and a double pointer, representing the circumference and array of sides, respectively
 - An integer, representing the number of sides
 - A default constructor, which will
 - Set the number of sides to a random value from 3 to 10
 - Create an array of that many sides and set them to random (double) values from 0 to 10
 - Calculate the circumference of the polygon for the set number and values of sides

Assignment 6

- A constructor with an integer parameter, which will
 - Set the number of sides to the value of the integer parameter
 - Create an array of that many sides and set them to random values (double) from 0 to 10
 - Calculate the circumference of the polygon for the set number and values of sides
- A constructor with an integer and a double parameter, which will
 - Set the number of sides to the value of the integer parameter
 - Create an array of that many sides and set them to random (double) values from 0 to the value of the double parameter
 - Calculate the circumference of the polygon for the set number and values of sides
- A constructor with a double parameter, which will
 - Set the number of sides to a random value from 3 to 10
 - Create an array of that many sides and set them to random (double) values from 0 to the value of the double parameter
 - Calculate the circumference of the polygon for the set

Assignment 6

- In the main() function:
 - Input the number of polygons (0 for a random value from 1 to 10)
 - Create (dynamically) an array of that many polygons
 - For each of the polygons in the array:
 - Enter the number of sides of that polygon; if 2 or less is entered, the number of sides of that polygon is assigned randomly
 - Enter the maximum length of any side of that polygon; if 0 or less is entered, the maximum length of the sides is assigned randomly
 - Initialize the current polygon using the appropriate constructor

Assignment 6

- After all of the polygons have been entered, find and store the index of the polygon with the greatest circumference, as well as the value of that circumference, and display

```
Polygon 0 has 4 sides:  
Length of side 0: 4.03516  
Length of side 1: 2.05756  
Length of side 2: 1.74657  
Length of side 3: 2.77261  
Its circumference is 10.6119
```

```
Polygon 1 has 4 sides:  
Length of side 0: 8.01233  
Length of side 1: 2.12012  
Length of side 2: 8.67763  
Length of side 3: 6.32313  
Its circumference is 25.1332
```

```
Polygon 2 has 4 sides:  
Length of side 0: 1.99591  
Length of side 1: 1.34709  
Length of side 2: 9.02036  
Length of side 3: 1.55126  
Its circumference is 13.9146
```

```
Polygon 3 has 3 sides:  
Length of side 0: 0.392695  
Length of side 1: 0.30498  
Length of side 2: 3.64502  
Its circumference is 4.3427
```

```
Polygon 4 has 4 sides:  
Length of side 0: 0.46839  
Length of side 1: 5.11694  
Length of side 2: 4.6188  
Length of side 3: 1.47028  
Its circumference is 19.4944
```

```
Polygon 1 has the greatest circumference of 25.1332  
Press any key to continue.
```

```
How many polygons will there be (0 for a random number)? 0  
There will be 5 polygons  
-----
```

```
Polygon 0:  
Enter the number of sides of the polygon (2 or less for a random number): 0  
Enter the maximum length of a side (0 or less for a random number): 0
```

```
Polygon 1:  
Enter the number of sides of the polygon (2 or less for a random number): 2  
Enter the maximum length of a side (0 or less for a random number): 0
```

```
Polygon 2:  
Enter the number of sides of the polygon (2 or less for a random number): 4  
Enter the maximum length of a side (0 or less for a random number): 0
```

```
Polygon 3:  
Enter the number of sides of the polygon (2 or less for a random number): 3  
Enter the maximum length of a side (0 or less for a random number): 7.789
```

```
Polygon 4:  
Enter the number of sides of the polygon (2 or less for a random number): 6  
Enter the maximum length of a side (0 or less for a random number): 5.9098
```