**Name:** Umar Farooq **Enrollment:**09-131242-088 **Section:**BSE-2B

# Lab # 3

# Classes using Constructor and Destructor

**Objective:**

The objective of this lab is to understand the concept of constructors and destructors in C++. By the end of this lab, you should be able to:

1. Define and use constructors to initialize objects.
2. Understand the difference between default and parameterized constructors.
3. Use destructors to clean up resources when an object is destroyed.
4. Apply constructors and destructors in real-world scenarios like creating a student database.

---

**1. Constructors**

- A constructor is a special member function of a class that is automatically called when an object of the class is created.
- It has the same name as the class and no return type (not even void).
- Constructors are used to initialize the data members of an object.
- Types of Constructors:
    - Default Constructor: A constructor with no parameters.
    - Parameterized Constructor: A constructor that takes parameters to initialize the object with specific values.

**2. Destructors**

- A destructor is a special member function of a class that is automatically called when an object goes out of scope or is explicitly deleted.
- It has the same name as the class preceded by a tilde (~).
- Destructors are used to release resources (e.g., memory, file handles) allocated by the object.

---

## Lab Activities:

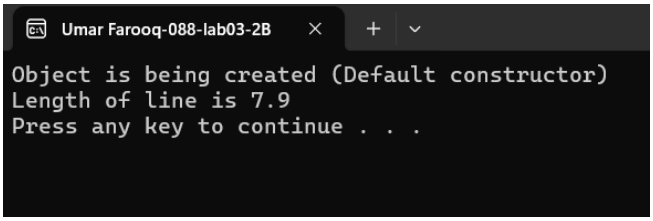
**Activity 1: Default Constructor**

**Code:**

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

class Line {
public:
   void setLength(double len);
   double getLength();
   Line(); // Default constructor

private:
   double length;
};

// Default constructor definition
Line::Line() {
   cout << "Object is being created (Default Constructor)" << endl;
   length = 0.0; // Initialize length to 0
}

void Line::setLength(double len) {
   length = len;
}

double Line::getLength() {
   return length;
}

int main() {
   Line line; // Object created, default constructor called

   // Set line length
   line.setLength(6.0);
   cout << "Length of line: " << line.getLength() << endl;

   return 0;
}
```

**Your Output Here….**

In case of output snippet please make sure output snippet contains student name and id. AliAhmed_123_Lab03_A1.exe




---

**Activity 2: Parameterized Constructor**

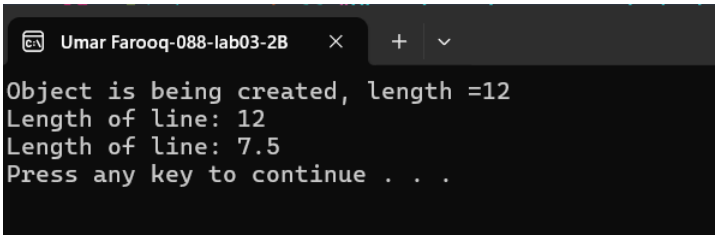
**Code:**

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

class Line {
public:
   void setLength(double len);
   double getLength();
   Line(double len); // Parameterized constructor

private:
   double length;
};

// Parameterized constructor definition
Line::Line(double len) {
   cout << "Object is being created, length = " << len << endl;
   length = len;
}

void Line::setLength(double len) {
   length = len;
}
```

```
double Line::getLength() {
   return length;
}

int main() {
   Line line(10.0); // Object created, parameterized constructor called

   // Get initially set length
   cout << "Length of line: " << line.getLength() << endl;

   // Set line length again
   line.setLength(6.0);
   cout << "Length of line: " << line.getLength() << endl;

   return 0;
}
```

**Your Output Here….**

In case of output snippet please make sure output snippet contains student name and id. `AliAhmed_123_Lab03_A2.exe`




---

**Activity 3: Student Database Using Constructor and Destructor**

Write a C++ program to create a student database using a class. The program should store the following details:

1. Name of the student
2. Roll number of the student
3. Height of the student
4. Weight of the student

Use a constructor to initialize the data members and a destructor to display a message when the object is destroyed.

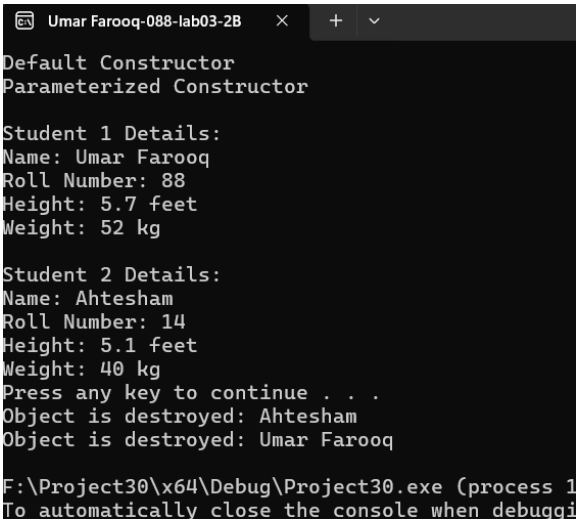
**Code:**

```
#include <iostream>
#include <cstring> // For strcpy
using namespace std;

class Student {
private:
    char name[25];
    int roll;
    float height, weight;

public:
    // Default constructor
    Student() {
        strcpy(name, "Ram");
        roll = 0;
        height = 0.0;
        weight = 0.0;
        cout << "Object created (Default Constructor)" << endl;
    }

    // Parameterized constructor
    Student(const char* n, int r, float h, float w) {
        strcpy(name, n);
        roll = r;
        height = h;
        weight = w;
        cout << "Object created (Parameterized Constructor)" << endl;
    }

    // Destructor
    ~Student() {
        cout << "Object destroyed for student: " << name << endl;
    }

    // Function to display student details
    void display() {
        cout << "\nName: " << name << endl;
        cout << "Roll No: " << roll << endl;
        cout << "Height: " << height << " feet" << endl;
        cout << "Weight: " << weight << " kg" << endl;
    }
};
```

```
int main() {
    // Create objects using default and parameterized constructors
    Student student1; // Default constructor
    Student student2("John", 101, 5.8, 65.5); // Parameterized constructor

    // Display student details
    cout << "\nStudent 1 Details:" << endl;
    student1.display();

    cout << "\nStudent 2 Details:" << endl;
    student2.display();

    return 0;
}
```

**Your Output Here….**

In case of output snippet please make sure output snippet contains student name and id. AliAhmed_123_Lab03_A3.exe




---

**Conclusion:**

In this lab, we learned:

1. How to define and use constructors to initialize objects.
2. The difference between default and parameterized constructors.
3. How to use destructors to clean up resources.
4. How to apply constructors and destructors in real-world scenarios like creating a student database.

---

**Additional Notes:**

- Constructors are essential for initializing objects with valid states.
- Destructors are crucial for releasing resources and preventing memory leaks.
- Always ensure that constructors and destructors are used appropriately in your programs.

# Lab Tasks/ Homework

**Task 01:** Write a class Result that contains the following:

- Data members: rollNo, name, and marks (an array of integers for three subjects marks).
- Member functions:
  - parameterized constructor to initialize rollNo, name, and marks.
  - destructor to display a message when the object is destroyed.
  - void input(); // input values for rollNo, name, and marks.
  - void show() ; // display the values of rollNo, name, and marks.
  - int total(); // calculate and return the total marks of the student.
  - float avg(); // calculate and return the average marks of the student.

**Solution:**

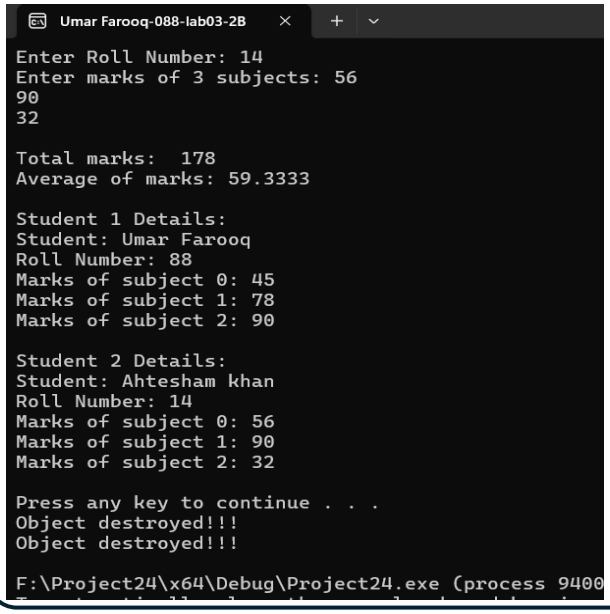
```
Enter your code here…
#include <iostream>
#include <string>
using namespace std;
class Student {
private:
	int marks[3];
	string name;
	int rollnumber;
public:
	void student(string n, int rn, int m1,int m2,int m3)
	{
		name = n;
		rollnumber = rn;
		marks[0] = m1;
		marks[1] = m2;
		marks[2] = m3;
	}
	~Student()
	{
		cout << "Object destroyed!!!" << endl;
	}
	void input()
	{
		cout << "Enter Student name: ";
		getline(cin, name);
		cout << "Enter Roll Number: ";
		cin >> rollnumber;
		cout << "Enter marks of 3 subjects: ";
		for (int i = 0; i < 3; i++)
		{
			cin >> marks[i];
```

```
            }
      }
      void show()
      {
            cout << "Student: " << name << endl;
            cout << "Roll Number: " << rollnumber << endl;
            for (int i = 0; i < 3 ; i++)
            {
                  cout << "Marks of subject " << i << ": " << marks[i]
<< endl;
            }
            cout << endl;
      }
      int total()
      {
            int sum;
            sum = marks[0] + marks[1] + marks[2];
            return sum;
      }
      float avg()
      {
            return total() / 3.0;
      }
};
int main()
{
      system("title Umar Farooq-088-lab03-2B");
      Student s1;
      s1.input();
      cout << endl;
      cout << "Total marks:  " << s1.total() << endl;
      cout << "Average of marks: " << s1.avg() << endl;
      cout << endl;


      Student s2;
      cin.ignore();
      s2.input();
      cout << endl;
      cout << "Total marks:  " << s2.total() << endl;
      cout << "Average of marks: " << s2.avg() << endl;
      cout << endl;
      cout << "Student 1 Details: " << endl;
      s1.show();
      cout << "Student 2 Details: " << endl;
      s2.show();
```

```
    system("pause");
}
```

**Output:**

Your Output here…




In case of output snippet please make sure output snippet contains student name and id. `AliAhmed_123_Lab03_T1.exe`

**Task 02:** Create a class Rectangle with the following:

- Data members: length and width (both default to 1).
- Member functions:
  - parameterized constructor to initialize length and width.
  - destructor to display a message when the object is destroyed.
  - void setLength(float l); // To set the length (verify that it is between 0.0 and 20.0).
  - void setWidth(float w); // To set the width (verify that it is between 0.0 and 20.0).
  - float getLength(): // To return the length.
  - float getWidth(): // To return the width.
  - float perimeter(): // To calculate and return the perimeter of the rectangle.
  - float area(): // To calculate and return the area of the rectangle.

**Solution:**

Enter your code here…

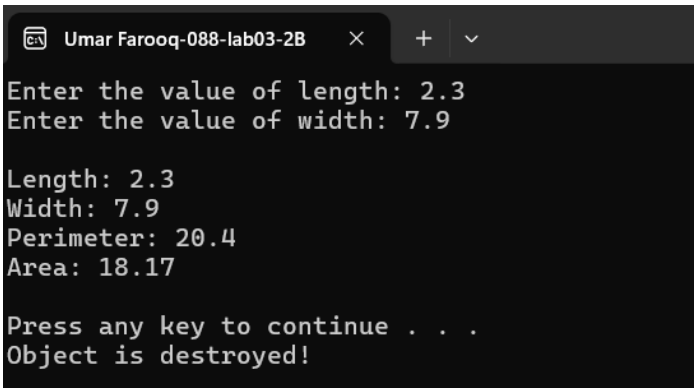
```
#include <iostream>
using namespace std;
class Rectangle {
private:
    float width;
    float length;
```

```
public:
	Rectangle() {
		length = 1;
		width = 1;
	}
	Rectangle(float w, float l)
	{
		width = w;
		length = l;
	}
	void setlength()
	{
		float len;
		cout << "Enter the value of length: ";
		cin >> len;
		if (len >= 0.0 && len <= 20.0)
		{
			length = len;
		}
		else
		{
			cout << "Invalid length!" << endl;
		}
	}
	void setwidth()
	{
		float wid;
		cout << "Enter the value of width: ";
		cin >> wid;
		if (wid >= 0.0 && wid <= 20.0)
		{
			width = wid;
		}
		else
		{
			cout << "Invalid Width!" << endl;
		}
	}
```

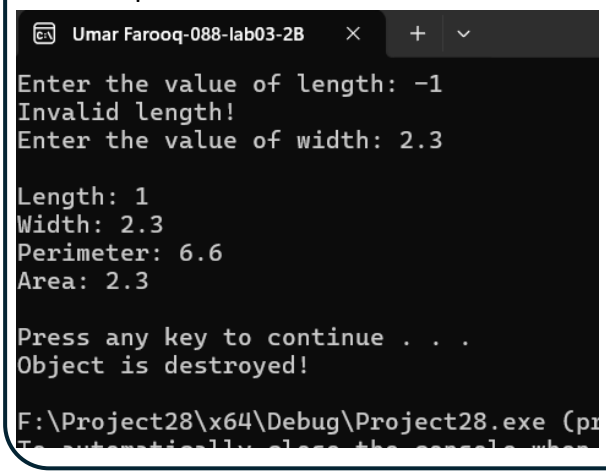
```
        float getlength()
        {
                return length;
        }
        float getwidth()
        {
                return width;
        }
        float perimeter()
        {
                return 2 * (length + width);
        }
        float area()
        {
                return length * width;
        }
        ~Rectangle()
        {
                cout << "Object is destroyed!" << endl;
        }
};
int main()
{
        system("title Umar Farooq-088-lab03-2B");
        Rectangle rect;
        rect.setlength();
        rect.setwidth();
        cout << endl;
        cout << "Length: " << rect.getlength() << endl;
        cout << "Width: " << rect.getwidth() << endl;
        cout << "Perimeter: " << rect.perimeter() << endl;
        cout << "Area: " << rect.area() << endl;
        cout << endl;
}
```

**Output:**

Your Output here…




Invalid input Screenshot:




In case of output snippet please make sure output snippet contains student name and id. `AliAhmed_123_Lab03_T2.exe`