Task 2: Utilize Constructors and Destructors (Car Class in C++)

Objective:
The objective of this task is to understand and implement constructors and destructors in C++. This task focuses on object initialization and destruction during program execution.

Problem Statement

Enhance the Car class by performing the following:
- Add a parameterized constructor to initialize make, model, and year
- Implement a destructor that displays a message when the object is destroyed

Concepts Used

- Constructor
- Destructor
- Object Lifecycle
- Encapsulation

Source Code (C++)

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

class Car {
private:
        string make;
        string model;
        int year;

public:
        // Constructor
        Car(string m, string mo, int y) {
        make = m;
        model = mo;
```

```
        year = y;
        cout << "Car object created using constructor." << endl;
        }

        // Getter functions
        string getMake() {
        return make;
        }

        string getModel() {
        return model;
        }

        int getYear() {
        return year;
        }

        // Destructor
        ~Car() {
        cout << "Car object destroyed using destructor." << endl;
        }
};

int main() {
        Car car1("Toyota", "Corolla", 2022);

        cout << "Car Make: " << car1.getMake() << endl;
        cout << "Car Model: " << car1.getModel() << endl;
        cout << "Car Year: " << car1.getYear() << endl;

        return 0;
}
```

Explanation of the Code

1. The Car class contains private data members to enforce encapsulation.
2. A parameterized constructor initializes all attributes when an object is created.
3. The constructor prints a message to indicate object creation.
4. Getter functions are used to access private data members safely.
5. The destructor is automatically called when the object goes out of scope.
6. The destructor prints a message indicating object destruction.

Program Output

Conclusion

This task successfully demonstrates the use of constructors and destructors in C++. Constructors ensure automatic initialization of objects, while destructors handle cleanup when objects are destroyed. These are core concepts in Object-Oriented Programming.