# Object Oriented Programming (OOP)
# Question Paper 2

**1. Write down the answers.**

**(a). What is a Destructor in Object Oriented Programming?**

**Ans.:** A destructor is an instance member function that is invoked automatically whenever an object is going to be destroyed. Meaning, a destructor is the last function that is going to be called before an object is destroyed.

A destructor has the same name as the class and is preceded by a **tilde (~)**. For example, the destructor for class `String` is declared: `~String()` .

**(b). Write down the differences between Class and Structure in C++.**

**Ans.:** Here are the differences between Class and Structure in C++:

| Class | Structure |
|---|---|
| Members of a class are **private** by default. | Members of a structure are **public** by default. |
| It is declared using the **class** keyword | It is declared using the **struct** keyword |
| It is normally used for data abstraction and inheritance. | It is normally used for the grouping of different datatypes. |

**2. Explain the inline function with a suitable example.**

**Ans.:** An inline function is a function that is expanded in line when it is called. We can use the `inline` keyword to create inline functions. Here is the general syntax:

```
inline returnType functionName(parameters) {
    // code
}
```

The use of inline functions can be demonstrated using the following example:

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

inline void display(int n) {
    cout << n << endl;
}

int main() {
    display(10);
    display(42);
    display(75);
    return 0;
}
```

Inline functions are copied straight to the location of the function call in compile-time to execute the program more efficiently.

**3. Write down a code for solving the given problem:**

- Implement a C++ class named **Room** that represents the dimensions of a room. The class should include **private** data members `length`, `height`, `width`, and provide a **parameterized constructor** for initializing these values. Additionally, you need to implement a **copy constructor** that duplicates the data of another `Room` object.
- The should have a method `calculateVolume()` to compute and display the room's area using the formula:

  ```
  length * height * width
  ```

- Create an instance of the `Room` class and initialize it with specific dimensions. Make a second object by copying the first object. Also, calculate and display the volumes of both `Room` objects.

**Ans.:** Here is a sample C++ program that satisfies the conditions in the given question:

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

```cpp
class Room
{
private:
    int length;
    int width;
    int height;

public:
    // parameterized constructor
    Room(int l, int w, int h)
    {
        length = l;
        width = w;
        height = h;
    }

    // copy constructor
    Room(const Room &r)
    {
        length = r.length;
        width = r.width;
        height = r.height;
    }

    void calculateVolume()
    {
        cout << length * width * height << endl;
    }
};

int main()
{
    Room r1(30, 40, 50);
    r1.calculateVolume();

    Room r2 = r1;
    r2.calculateVolume();
    return 0;
}
```

# References


- **Microsoft Learn**: Destructors in C++
- **GeeksforGeeks**: Destructors in C++
- **GeeksforGeeks**: Difference Between Structure and Class in C++
- **Programiz**: C++ Inline Functions