

**Experiment No-03:** Static Data Member, and Function Overloading in C++.

**Objectives**

- Introduce with the Static Data Member and Member function.
- Understand the concept of function overloading in C++.

✓ **Example 1:** A C++ program to demonstrate the use of static data member.

**Static Data member: [objectCount]**

---

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

class Square {
private:
    int side; // normal data member
public:
    static int objectCount; // static data member

    // Constructor definition
    Square()
    {
        // Increase every time object is created
        objectCount++;
    }
};

// Initialize static member of class Square using scope resolution
operator

int Square::objectCount = 0;

int main() {
    Square s1;
    // Object Count.
    cout << "Total objects: " << Square::objectCount << endl;
    Square s2;
    // Object Count.
    cout << "Total objects: " << Square::objectCount << endl;
    return 0;
}
```

---

**Example 2:** A C++ program to demonstrate the use of static member function.

**Static Data member: [objectCount]**

---

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

```
class Square {
    private:
        int side; // normal data member
        static int objectCount; // static data member
    public:

        // Constructor definition
        Square()
        {
            // Increase every time object is created
            objectCount++;
        }

        // creating a static function that returns static data member
        static int getCount() {
            return objectCount;
        }
};

// Initialize static member of class Square
int Square::objectCount = 0;

int main() {
    Square s1;
    // Object Count.
    cout << "Total objects: " << Square::getCount() << endl;
    Square s2;
    // Object Count.
    cout << "Total objects: " << Square::getCount() << endl;
    return 0;
}
```

---

**Example 3:** A program to understand the Function Overloading in C++.

---

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

void print(int var) {
    cout << "Integer number: " << var << endl;
}

void print(float var) {
    cout << "Float number: " << var << endl;
}

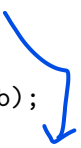
void print(int var1, float var2) {
    cout << "Integer number: " << var1;
    cout << " and float number:" << var2;
}

int main() {
```

```
int a = 4;
float b = 3.5;

print(a);
print(b);
print(a, b);

return 0;
}
```



---

\*\*\* For better understanding please feel free to search on internet because it is the best source of learning. \*\*\*

### Practice Exercise

1. Write a C++ program to define a class **Batsman** with the following specifications,

**batsman\_ID:** 6 digits roll number

**static member count:** To keep track on number of object

**static function getcount():** return the value of count

**function getname():** To take batsman name as input

**function showname():** To show batsman name

Access all the data members and member functions using the objects of class Batsman.

### Sample Input/Output

Initially number of objects: 0

Enter number of entry: 2

Enter Batsman Name: Shakib

Enter Batsman Name: Liton

Finally number of objects: 2

Batsman Name: Shakib

Batsman Name: Liton

2. Write a C++ Program to calculate the area of different geometric shapes such as Circle, Triangle, and Rectangle. Use function overloading.

**Class Name: Shape**

[\[Resource Link 1\]](#)  
[\[Resource Link 2\]](#)