1- Write a program that uses a static variable in a function. The program must call the function several times, at least once incrementing the variable by one and displaying the value of the constant variable.

اكتب برنامجًا يستخدم static variable in a function. يجب أن يقوم البرنامج باستدعاء function عدة مرات، على الأقل مرة واحدة مع زيادة المتغير بمقدار واحد وعرض قيمة المتغير الثابت.

Output

```
Calling the function multiple times:
Static Variable Value: 1
Static Variable Value: 2
Static Variable Value: 3
```

Solution

```cpp
// www.gammal.tech

#include<iostream>
using namespace std;

void demo() {
    static int count = 0;
    cout << "Static Variable Value: " << ++count << endl;
}

int main() {
    cout << "Calling the function multiple times:" << endl;
    demo();
    demo();
    demo();

    return 0;
}
```

2- create a C++ program with a class MyClass that includes a static member variable count. The class should have a constructor that increments the count every time an object is created. In the main function, create three objects of the class and print the total number of objects created.

إنشاء برنامج with a class MyClass that includes a static member variable count.
class should have a constructor في كل مرة يتم فيها إنشاء object .
في main function، قم بإنشاء ثلاثة object من class وطباعة العدد الإجمالي objects التي تم إنشاؤها.

Output

```
Number of Objects: 3
```

Solution

```cpp
// www.gammal.tech

#include<iostream>
using namespace std;

class MyClass {
public:
    static int count;

    MyClass() {
        count++;
    }
};

int MyClass::count = 0;

int main() {
    MyClass obj1, obj2, obj3;
    cout << "Number of Objects: " << MyClass::count << endl;

    return 0;
}
```

3- Create a program that includes a class MyClass with a static member function named staticFunction. In the main function, call the static member function without creating an object of the class.

Output

```
Static Member Function
```

Solution

```cpp
// www.gammal.tech

#include<iostream>
using namespace std;

class MyClass {
public:
    static void staticFunction() {
        cout << "Static Member Function" << endl;
    }
};

int main() {
    MyClass::staticFunction();

    return 0;
}
```

4- Create a program that includes a function printArray which contains a static array. In the main function, call the printArray function twice and observe the behavior of the static array.

قم بإنشاء برنامج يتضمن function printArray التي تحتوي على مصفوفة ثابتة. في الوظيفة الرئيسية، قم باستدعاء function printArray مرتين ولاحظ سلوك array الثابتة.

Output

```
Calling the function with static array:
Static Array Elements: 1 2 3 4 5
Static Array Elements: 1 2 3 4 5
```

Solution

```cpp
// www.gammal.tech

#include<iostream>
using namespace std;

void printArray() {
    static int arr[5] = {1, 2, 3, 4, 5};
    cout << "Static Array Elements: ";
    for (int i = 0; i < 5; ++i) {
        cout << arr[i] << " ";
    }
    cout << endl;
}

int main() {
    cout << "Calling the function with static array:" << endl;
    printArray();
    printArray();

    return 0;
}
```

5- Create a program that includes a function calculateSum which calculates the cumulative sum of user-inputted numbers using a static variable. In the main function, call the calculateSum function three times and observe the cumulative sum.

أنشئ برنامجًا يتضمن function calculateSum التي تحسب المجموع التراكمي للأرقام التي يدخلها المستخدم باستخدام متغير ثابت. في الوظيفة الرئيسية، قم باستدعاء function calculateSum ثلاث مرات ولاحظ المجموع التراكمي.

Input & Output

```
Calculating cumulative sum with user input:
Enter a number: 3
Cumulative Sum: 3
Enter a number: 5
Cumulative Sum: 8
Enter a number: 6
Cumulative Sum: 14
```

Solution

```cpp
// www.gammal.tech

#include<iostream>
using namespace std;

void calculateSum() {
    static int sum = 0;
    int num;
    cout << "Enter a number: ";
    cin >> num;
    sum += num;
    cout << "Cumulative Sum: " << sum << endl;
}

int main() {
    cout << "Calculating cumulative sum with user input:" << endl;
    calculateSum();
    calculateSum();
    calculateSum();

    return 0;
}
```

6- create a program that includes a function demoFunction with a static variable.and increment of +1 each time , In the main function, call demoFunction in a loop and observe the behavior of the static variable.

إنشاء برنامج يتضمن function demoFunction with a static variable. وزيادة +1 في كل مرة, في الوظيفة الرئيسية، call demoFunction in a loop ولاحظ سلوك المتغير الثابت.

Output

```
Static Variable Value: 0
Static Variable Value: 1
Static Variable Value: 2
Static Variable Value: 3
Static Variable Value: 4
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

void demoFunction() {
    static int staticVar = 0;
    cout << "Static Variable Value: " << staticVar << endl;
    staticVar++;
}

int main() {
    for (int i = 0; i < 5; ++i) {
        demoFunction();
    }
    return 0;
}
```

7- Create a program that includes a function demoFunction with a static variable. In the main function, call demoFunction with both true and false conditions and observe the behavior of the static variable.

قم بإنشاء برنامج يتضمن function demoFunction with a static variable. في الوظيفة الرئيسية، In the main function, call demoFunction مع كل من الشرط الصحيح والخاطئ ولاحظ سلوك المتغير الثابت.

Output

```
Static Variable Value: 10
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

void demoFunction(bool condition) {
    if (condition) {
        static int staticVar = 10;
        cout << "Static Variable Value: " << staticVar << endl;
    }
}

int main() {
    demoFunction(true);
    demoFunction(false);
    return 0;
}
```

8- Create a program that includes a trial function with a static variable and a local variable. Increment each variable once ,in the main function, call the demoFunction several times and observe the behavior of both variables.

إنشاء برنامج يتضمن trial function with a static variable and a local variable. قم بزيادة كل متغير مرة واحدة، في الوظيفة الرئيسية، قم باستدعاء trial function عدة مرات ولاحظ سلوك كلا المتغيرين.

Output

```
Static Variable Value: 0
Local Variable Value: 0
Static Variable Value: 1
Local Variable Value: 0
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

void demoFunction() {
    static int staticVar = 0;
    int localVar = 0;
    cout << "Static Variable Value: " << staticVar << endl;
    cout << "Local Variable Value: " << localVar << endl;
    staticVar++;
    localVar++;
}

int main() {
    demoFunction();
    demoFunction();
    return 0;
}
```

9- Create a program that uses an enumeration Color to represent different colors. Initialize a static variable selectedColor with the value GREEN and print its value.

{ RED, GREEN, BLUE }

قم بإنشاء برنامج enumeration Color التعداد لتمثيل ألوان مختلفة.
Initialize a static variable اللون المحدد بالقيمة GREEN وطباعة قيمته.

{ RED, GREEN, BLUE }

Output

```
Selected Color: 1
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

enum Color { RED, GREEN, BLUE };

int main() {
    static Color selectedColor = GREEN;
    cout << "Selected Color: " << selectedColor << endl;
    return 0;
}
```

10- Create a program with a class GT_Bank_Account to represent a bank account. The class has a static variable balance to store the account balance. Implement two derived classes sales and bills from the base class GT_Bank_Account. The sales class should have a method deposit to add money to the balance, and the bills class should have a method withdraw to subtract money from the balance. Additionally, create another derived class accountant with a method getBalance to display the current balance. Demonstrate the usage of these classes in the main function by creating objects, making deposits and withdrawals, and printing the final balance.

Output

```
Balance: 3550
```

Solution

```cpp
// www.gammal.tech

#include <iostream>
using namespace std;

class GT_Bank_Account {
public:
    static double balance; // Reserve a variable for the balance
};

class sales : public GT_Bank_Account {
public:
    void deposit(double amount) {
        balance += amount;
    }
};

class bills : public GT_Bank_Account {
public:
    void withdraw(double amount) {
        balance -= amount;
    }
};

class accountant : public GT_Bank_Account {
public:
    void getBalance() {
        cout << "Balance: " << balance << endl;
    }
};

double GT_Bank_Account::balance = 0;

int main() {
    sales s;
    s.deposit(100);
    s.deposit(3500);

    bills b;
    b.withdraw(50);

    accountant a;
    a.getBalance();

    return 0;
}
```