Assignment 04 STATIC

1. Create a class Book with members as bid,bname,price and author.Add following methods:
   a. Constructor (Support both parameterized and parameterless)
   b. Destructor
   c. ShowBook
   d. Add static variable count and also maintain count of objects created.

Book.h

```
// 1. Create a class Book with members as bid,bname,price and author.Add following
methods:
#include <bits/stdc++.h>
using namespace std;
class Book
{
private:
    int id;
    string bname;
    string author;
    double price;
    static int count;
    static string lib_name;

public:
    Book();
    Book(int, string, string, double);
    void displayBook();

    void static showLibName();
    void static showCount();
    void static updateLibName(string);
    // ~Book();
};
```

Book.cpp

```
#include "book.h"
string Book::lib_name = "My Library";
int Book::count = 0;
Book::Book()
{
    this->author = "Not given";
    this->bname = "Not given";
    this->price = 0.0;
    this->id = 0;
    count++;
}
Book::Book(int id, string name, string author, double price)
{
```

```
    count++;
    this->author = author;
    this->bname = name;
    this->price = price;
    this->id = id;
}

void Book::displayBook()
{
    cout << "\nBook Id     : " << this->id;
    cout << "\nBook Name   : " << this->bname;
    cout << "\nBook Author : " << this->author;
    cout << "\nBook Price  : " << this->price;
}

void Book::showLibName()
{
    cout << "\nLibrary Name :" << lib_name;
}

void Book::updateLibName(string libname)
{
    lib_name = libname;
}

void Book::showCount()
{
    cout << "\nObject Count :" << count;
}
```

Main.cpp

```
#include "book.h"

int main()
{
    Book b1(1, "Ikigai", "Byrne", 149.99);
    b1.displayBook();
    Book::showCount();
    cout << "\n";
    Book b2;
    b2.displayBook();
    Book::showCount();
    cout << "\n";

    Book b3(2, "The Secret", "Rhonda", 199.99);
    b3.displayBook();
    Book::showCount();

    cout << "\n";
```

```
    Book::showLibName();
    Book::updateLibName("The World Library");
    cout << "\nUpdated : ";
    Book::showLibName();
    return 0;
}
```

OUTPUT:

PS D:\Fullstack-Java-FirstBit-Solutions\DSA\Assignments\Static\Book> ./main

Book Id   : 1

Book Name   : Ikigai

Book Author : Byrne

Book Price : 149.99

Object Count :1

Book Id   : 0

Book Name   : Not given

Book Author : Not given

Book Price : 0

Object Count :2

Book Id   : 2

Book Name   : The Secret

Book Author : Rhonda

Book Price : 199.99

Object Count :3

Library Name :My Library

Updated :

Library Name :The World Library

PS D:\Fullstack-Java-FirstBit-Solutions\DSA\Assignments\Static\Book>

2. Create a class Product with members as pid,pname,price and quantity .Add following methods:
    e. Constructor (Support both parameterized and parameterless)
    f. Destructor
    g. ShowBook
    h. Add static member discount.
    i. Provide methods for applying discount on price of product.

Product.h

```
#include <bits/stdc++.h>

using namespace std;

class Product
{
private:
    int pid, quantity;
    string pname;
    double price;
    static int prodCount;

public:
    Product();
    Product(int, string, int, double);
    void showProduct();
    static void showProdCount();
};
```

Product.cpp

```
#include "product.h"
int Product::prodCount = 0;

Product::Product()
{
    this->pid = 0;
    this->pname = "Not Given";
    this->price = 0.0;
    this->quantity = 0;
    prodCount++;
}
Product::Product(int id, string name, int quantity, double price)
{
    this->pid = id;
    this->pname = name;
    this->price = price;
    this->quantity = quantity;
    prodCount++;
```

```
}
void Product::showProduct()
{
    cout << "\nProduct ID       : " << this->pid;
    cout << "\nProduct Name     : " << this->pname;
    cout << "\nProduct Price    : " << this->price;
    cout << "\nProduct Quantity : " << this->quantity;
}
void Product::showProdCount()
{
    cout << "\nProduct Count    : " << prodCount;
}
```

Main.cpp

```
#include "product.h"

int main()
{
    Product::showProdCount();
    Product b1(1, "Mobile", 23, 1499.99);
    b1.showProduct();
    Product::showProdCount();

    cout << "\n";
    Product b2;
    b2.showProduct();
    Product::showProdCount();

    cout << "\n";
    Product b3(2, "Laptop", 12, 3455.67);
    b3.showProduct();
    Product::showProdCount();
    return 0;
}
```

Output: PS D:\Fullstack-Java-FirstBit-Solutions\DSA\Assignments\Static\Product> ./main

Product Count   : 0

Product ID   : 1

Product Name   : Mobile

Product Price   : 1499.99

Product Quantity : 23

Product Count   : 1

Product ID   : 0

Product Name   : Not Given

Product Price : 0

Product Quantity : 0

Product Count : 2

Product ID : 2

Product Name : Laptop

Product Price : 3455.67

Product Quantity : 12

Product Count : 3

PS D:\Fullstack-Java-FirstBit-Solutions\DSA\Assignments\Static\Product>

3. Create a class Shirt with members as sid,sname,type(formal etc), price and size(small,large etc) .Add following methods:

j. Constructor (Support both parameterized and parameterless)

k. Destructor

l. ShowBook

m. For each size of shirt price should change by 10%.
(eg. If 1000 is price then small price = 1000, medium = 1100,large=1200 and xlarge=1300) Use static concept

Shirt.h

```
#include <bits/stdc++.h>
using namespace std;

class Shirt
{
private:
    int sid;
    double price;
    string sname, sType, sSize;
    static int prodCount;

public:
    // Constructors
    Shirt();
    Shirt(int, string, string, string, double);

    // Methods
    void displayShirt();
    static double calculateFinalPrice(const string& size, double basePrice); // Adjust
price by size

    // Static Method for Object Count
    static int getProdCount() { return prodCount; }
};
```

Shirt.cpp

```
#include "shirt.h"

int Shirt::prodCount = 0;


Shirt::Shirt()
{
    this->sid = 0;
    this->price = 0.0;
    this->sname = "Not Given";
    this->sSize = "Not Given";
```

```
    this->sType = "Not Given";
    prodCount++;
}

Shirt::Shirt(int id, string name, string type, string size, double price)
{
    this->sid = id;
    this->price = price;
    this->sname = name;
    this->sSize = size;
    this->sType = type;
    prodCount++;
}

void Shirt::displayShirt()
{
    cout << "\nId    : " << this->sid;
    cout << "\nPrice : " << this->price;
    cout << "\nName  : " << this->sname;
    cout << "\nSize  : " << this->sSize;
    cout << "\nType  : " << this->sType;
    cout << "\nFinal Price (Adjusted) : " << calculateFinalPrice(sSize, price) << endl;
}

double Shirt::calculateFinalPrice(const string &size, double basePrice)
{
    if (size == "Medium")
        return basePrice * 1.1;
    else if (size == "Large")
        return basePrice * 1.2;
    else if (size == "X-Large")
        return basePrice * 1.3;
    else
        return basePrice;
}
```

Main.cpp

```
#include "shirt.h"

int main()
{

    Shirt shirt1(1, "Formal Shirt", "Formal", "Medium", 1000);
    Shirt shirt2(2, "Casual Shirt", "Casual", "Large", 1200);
    Shirt shirt3(3, "Party Shirt", "Party Wear", "Small", 1500);

    cout << "Displaying Shirt Details:\n";
    shirt1.displayShirt();
    shirt2.displayShirt();
    shirt3.displayShirt();
```

```
    cout << "\nTotal Products Created: " << Shirt::getProdCount() << endl;

    return 0;
}
```

Output: PS D:\Fullstack-Java-FirstBit-Solutions\DSA\Assignments\Static\Shirt> ./main

Displaying Shirt Details:

Id : 1

Price : 1000

Name : Formal Shirt

Size : Medium

Type : Formal

Final Price (Adjusted) : 1100

Id : 2

Price : 1200

Name : Casual Shirt

Size : Large

Type : Casual

Final Price (Adjusted) : 1440

Id : 3

Price : 1500

Name : Party Shirt

Size : Small

Type : Party Wear

Final Price (Adjusted) : 1500

Total Products Created: 3

PS D:\Fullstack-Java-FirstBit-Solutions\DSA\Assignments\Static\Shirt>