



**Bahria University, Islamabad**

**Department of Software Engineering**

**Object Oriented Programming Lab (Spring-2025)**

**Teacher: Engr. Muhammad Faisal Zia**

---

**Student: Umar Farooq**

**Enrollment: 09-131242-088**

---

**Lab Journal: 4**

**Date: 08/03/2025**

---

| Task No: | Task Wise Marks |          | Documentation Marks |          | Total Marks (20) |
|----------|-----------------|----------|---------------------|----------|------------------|
|          | Assigned        | Obtained | Assigned            | Obtained |                  |
| 1        | 2               |          | 4                   |          |                  |
| 2        | 2               |          |                     |          |                  |
| 3        | 2               |          |                     |          |                  |
| 4        | 5               |          |                     |          |                  |
| 5        | 5               |          |                     |          |                  |

**Comments:**

**Signature**

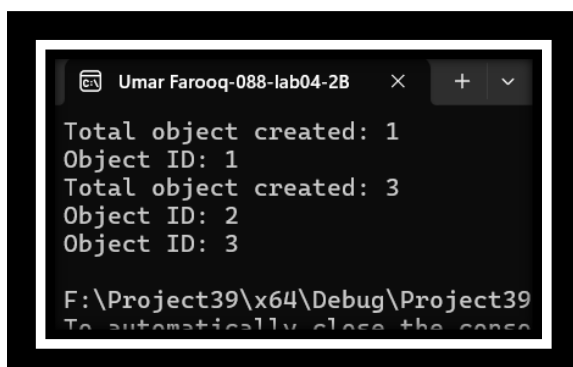
## Lab Activities:

### Activity 1: Object Counting Using Static Variable

#### Code:

```
#include <iostream>
using namespace std;
class Alpha {
private:
    static int total;
    int ID;
public:
    Alpha()
    {
        total++;
        ID = total;
    }
    static void showtotal()
    {
        cout << "Total object created: " << total << endl;
    }
    void showID()
    {
        cout << "Object ID: " << ID << endl;
    }
};
int Alpha::total = 0;
int main()
{
    Alpha obj1;
    Alpha::showtotal();
    obj1.showID();
    Alpha obj2, obj3;
    Alpha::showtotal();
    obj2.showID();
    obj3.showID();
    return 0;
}
```

#### Screenshot:



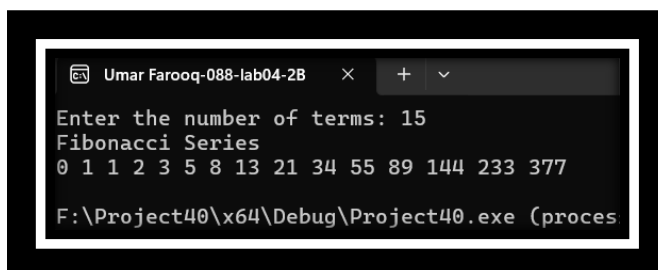
### Activity 2: Fibonacci Series Using Static Variable

Write a program to print the Fibonacci series using a static variable.

#### Code:

```
#include <iostream>
using namespace std;
class Fibonacci {
private:
    static int a, b;
public:
    static void Generate(int n)
    {
        for (int i = 0; i < n; i++)
        {
            cout << a << " ";
            int next = a + b;
            a = b;
            b = next;
        }
        cout << endl;
    }
};
int Fibonacci::a = 0;
int Fibonacci::b = 1;
int main()
{
    int n;
    cout << "Enter the number of terms: ";
    cin >> n;
    cout << "Fibonacci Series " << endl;
    Fibonacci::Generate(n);
    return 0;
}
```

#### Screenshot:



## Home Tasks

**Task 01:** Create a class SavingsAccount that uses a static data member to store the annualInterestRate for all savers. Each object of the class should contain a private data member

savingBalance to store the amount the saver currently has deposited. Provide the following member functions:

1. A constructor to initialize the savingBalance.
2. A static function setAnnualInterestRate() to set the annualInterestRate.
3. A member function monthlyInterest() to calculate the monthly interest by multiplying the savingBalance by annualInterestRate divided by 12.
4. A member function displayBalance() to display the current balance after adding the monthly interest.

**Code:**

```
#include <iostream>
using namespace std;
class SavingAccount {
private:
    double savingBalance;
    static double annualIntrestRate;
public:
    SavingAccount(double balance)
    {
        savingBalance = balance;
    }
    static void setannualIntrestRate(double rate)
    {
        annualIntrestRate = rate;
    }
    void monthlyIntrest()
    {
        double intrest = (savingBalance * annualIntrestRate) / 12;
        savingBalance += intrest;
    }
    void Displaybalance()
    {
        cout << "Current Balance: $" << savingBalance << endl;
    }
};
double SavingAccount::annualIntrestRate = 0.0;
int main()
{
    SavingAccount s1(20500.0);
    SavingAccount s2(3370.0);
    SavingAccount::setannualIntrestRate(0.08);
    cout << "Saver 1:" << endl;
    s1.monthlyIntrest();
    s1.Displaybalance();

    cout << "Saver 2:" << endl;
    s1.monthlyIntrest();
    s2.Displaybalance();
}
```

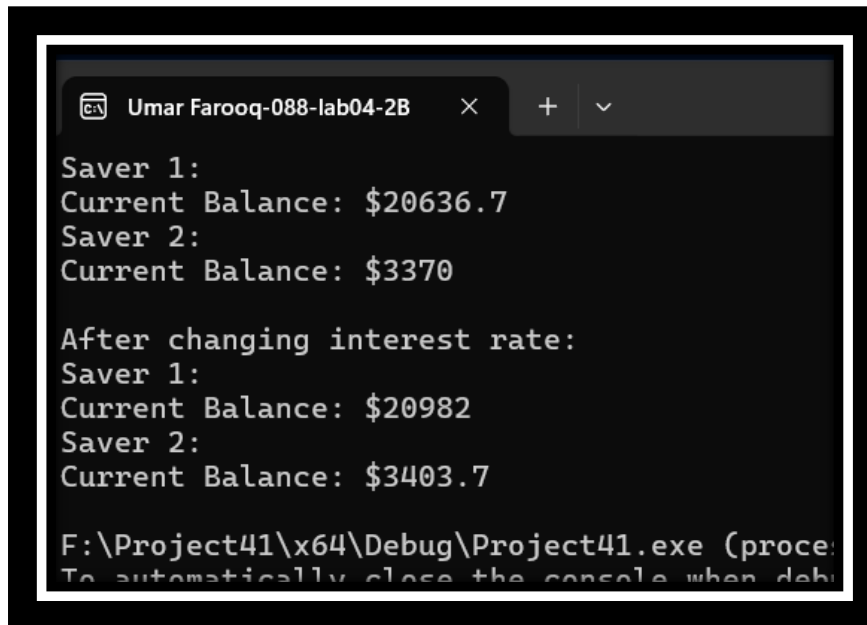
```
        cout << endl;
        SavingAccount::setannualIntrestRate(0.12);

        cout << "After changing interest rate:" << endl;
        cout << "Saver 1:" << endl;
        s1.monthlyIntrest();
        s1.Displaybalance();

        cout << "Saver 2:" << endl;
        s2.monthlyIntrest();
        s2.Displaybalance();

        return 0;
    }
```

### Screenshot:



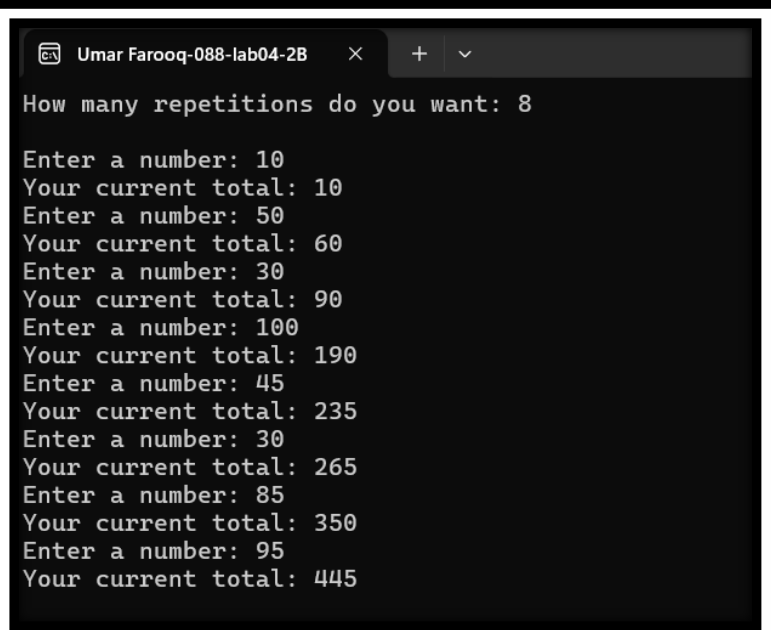
**Task 02:** Write a program that sums integers, using static variables.

### Code:

```
#include <iostream>
using namespace std;
class sum {
private:
    static int total;
public:
    void addNumber(int num)
    {
        total += num;
        cout << "Your current total: " << total << endl;
    }
};
int sum::total = 0;
```

```
int main()
{
    sum calculate;
    int num, rep;
    cout << "How many repetitions do you want: ";
    cin >> rep;
    cout << endl;
    for (int i = 0; i < rep; i++)
    {
        cout << "Enter a number: ";
        cin >> num;
        calculate.addNumber(num);
    }
    return 0;
}
```

### Screenshot:



The screenshot shows a terminal window titled "Umar Farooq-088-lab04-2B". The program prompts the user for the number of repetitions, which is 8. It then enters a loop where it repeatedly asks for a number to be added to a running total. The output shows the current total after each input.

```
Umar Farooq-088-lab04-2B
How many repetitions do you want: 8
Enter a number: 10
Your current total: 10
Enter a number: 50
Your current total: 60
Enter a number: 30
Your current total: 90
Enter a number: 100
Your current total: 190
Enter a number: 45
Your current total: 235
Enter a number: 30
Your current total: 265
Enter a number: 85
Your current total: 350
Enter a number: 95
Your current total: 445
```