Static Class Data and Static Member Function

1. Write a program to count the number of objects created for any class using static data member.

2. Create a SavingsAccount class. Use a static data member annualInterestRate to store the annual interest rate. The class contains a private data member savingsBalance indicating the balance of account. Provide member function calculateMonthlyInterest that calculates the monthly interest by multiplying the balance by annualInterestRate divided by 12; this interest should be added to savingsBalance.
   Write a driver program to test class SavingsAccount. Instantiate two different objects of class SavingsAccount, saver1 and saver2, with balances of $2000.00 and $3000.00, respectively. Set the annualInterestRate to 3 percent. Then calculate the monthly interest and print the new balances for each of the savers. Then set the annualInterestRate to 4 percent, calculate the next month's interest and print the new balances for each of the savers.

3. Create a Calculator class that has following methods:

   sum, multiply, divide , modulus , sin , cos , tan

   The user should be able to call these methods without creating an object of Calculator class.

Immutable Classes

1. Convert the following class to immutable class:

```
public class Fraction {
   int numerator, denominator;

   public Fraction(int numerator, int denominator) {
      this.numerator = numerator;
      this.denominator = denominator;
   }
```

```
   public int getNumerator() {
      return numerator;
   }

   public void setNumerator(int numerator) {
      this.numerator = numerator;
   }

   public int getDenominator() {
      return denominator;
   }

   public void setDenominator(int denominator) {
      this.denominator = denominator;
   }


         public void add(Fraction other) {
   numerator = numerator * other.denominator
         + other.numerator * denominator;
   denominator = denominator * other.denominator;
}

}
```