Task 1:

Let student information system of a school keeps the records of every student's name, date of birth, id and their respective cgpa. Create a class called student that will allow you to store all these information regarding a student.
You should define setInfo() function which will set the necessary information of a student object.
Define a function named getInfo() which will return all the stored information belonging a student object.
Include a constructor function to initialize the student object by zero and null values.
All the member variable to student class should be private.

Task 2:
Include one more **const** member function named **getAge**() which will return the age of a student calculated from date of birth. To accomplish this task you have to take additional user input to take current date.
Your program should check the validity (whether the date is valid or not) of date of birth and current date.

Task 3:
Write an overloaded function **getAge()** [Same task as task 2 but ..] by **taking system time** instead of your input for current date. You can take the help from any reference book to check how to take system time in C++.

Task 4:

Create a **SavingsAccount** class. Use a **static data** member **annualInterestRate** to store the annual interest rate for each of the savers. Each member of the class contains a private data member **savingsBalance** indicating the **amount the saver currently has on deposit**. 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**.

Provide a **static member function** modifyInterestRate that sets the static annualInterestRate to a new value.

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. **Also, count the number of objects created and destroyed for a class using static data members and static member functions**