Create a Time class that has three private data attributes: hour, minute, and second. Define appropriate setters and getters for these attributes. Write a public display() to display the time in this format: hh:mm:ss. It has a constructor that takes 3 integer arguments to initialize its three data attributes.

Create a SavingsAccount class that has a private data attribute annualInterestRate that is to store the annual interest rate for each of the savers. You will need to write an appropriate constructor to initialize this attribute, and other attributes described in next paragraph.

The class also contains a private data attribute savingsBalance indicating the amount the saver currently has on deposit, as well as a data attribute updatedTime indicating what time the balance was updated. The data type of updatedTime is Time class. Provide member method calculateMonthlyInterest that calculates the monthly interest by multiplying the balance by annualInterestRate divided by 12; this interest should be added to savingsBalance, and the updatedTime should be updated to the current time.

Both Time class and SavingsAccount class will be define in one Python source file as a module.

Create another Python source file as a test program to test class SavingsAccount. Instantiate two different objects of class SavingsAccount, saver1 and saver2, with balances of $5000.00 and $10000.00, respectively. Set the annualInterestRate to 1.5 percent. Then calculate the monthly interest and print the new balances for each of the savers. Then set the annualInterestRate to 3 percent, calculate the next month's interest and print the new balances for each of the savers.

Sample of Execution:
Initial balances:
Saver 1: $5000.00, updated: 12:22:88
Saver 2: $10000.00, updated: 12:22:88

Balances after 1 month's interest applied at .015:
Saver 1: $5006.25, updated: 12:25:88
Saver 2: $10012.50, updated: 12:25:88

Balances after 1 month's interest applied at .03:
Saver 1: $5018.77, updated: 12:35:88
Saver 2: $10037.53, updated: 12:35:88