1-

Create a class **SavingAccount** that will be used to instantiate bank account objects.

The class should have the following data members:

- **`AcctNbr`** - The Account Number
  (Between 1000 and 10000. Defaults to **0**)
- **`annualInterestRate`** - The annual interest rate
  (Between 0.0 and 100.0. Defaults to **0.0**)
- **`savingBalance`** - The Account's balance
  (Greater or equal to 0.0 Defaults to **0.0**)

and the following member functions:

- A **`constructor`** with no arguments.
- An **`overloaded constructor`** that takes as arguments, the values to set the data members.
- All needed **`setter`** and **`getter`** functions.
- A member function **`crdTrans`** that takes as an argument a value of type double. This value will be ***added*** to the ***account balance***.
- A member function **`dbtTrans`** that takes as an argument a value of type double. This value will be ***subtracted*** from the ***account balance***.

Write a program to test your class by instantiating object and accessing members through this object.

Include in your program, two functions that takes as argument a reference to an object of your class. One function calculates and returns the annual interest and the second function calculates and returns the monthly interest

Annual interest = balance * rate / 100

Monthly interest = Annual interest / 12

2-

Modify your class **SavingAccount** by allowing the external function that calculates the ***annual interest*** to access all the classe's members (private and public)

3-

Modify your test program by asking the user to enter the number of accounts (object) he needs to create.
Your program should create dynamic objects and manage their use.

4-

Modify your class by adding data memebers to better manage the dynamicaly allocated object. (number of object available, next object to be used, etc…)