When asked about the wonders of the world, Albert Einstein had an interesting answer – "Compound interest is the eighth wonder of the world. He who understands it, earns it ... he who doesn't ... pays it." Your assignment will help you better understand the significance of this observation.

Sound familiar?

Well it is, this time however, your program, Assignment5.java will use the following techniques:

- An **enum** (ref: p 335, Section 8.9 **enum Types**) named **Family** for the following four individuals:
    - Wilma & Fred Flintsone, Betty & Barney Rubble. (Those are their first and last names.)
    - Each enum will have the following data:
        - First name
        - Last name
        - Annual deposit
        - Annual interest rate – in decimal percentage, that is 0.05 represents 5%, the data would 0.05 NOT 5.
        - Number of years to deposit
        - Number of years to calculate compound interest
            - HINT: The data shown above will "properties" for each specific enum, see next page.
    - These enums, named WILMA, FRED, BETTY, & BARNEY will provide the data to calculate their retirement savings using **BigDecimal** data types (ref p 345, Section 8.15 **Using BigDecimal for Precise Monetary Calculations**) for the money calculations.
    - There should be an array of annual deposits based on the properties of the enum, i.e. annual deposit, number of years to deposit, etc.

Your program, named Assignment5.java, has the following requirements:

- Display the "current year" for the calculations described below.
- Calculate the annual interest using a **BigDecimal** containing the combined interest and principal for the given year.
- Display the cumulative annual deposits – for each member of the **FAMILY.**
- Display the running total value of the accumulated principal and interest.

GRADING:

The total possible score for this program is 100 points.

The following point values will be deducted for the reasons stated:

-100 points The program does not successfully compile from the command line

-90 points The program does not run from the command line without error or produces no output.

-70 points The program compiles, runs, and crashes thereafter or produces no output.

-50 points The program compiles, runs, collects the inputs, and produces no output. The program does NOT contain the specified enum, **Family.** The program does NOT use **BigDecimal** for the financial calculations.

-25 points The program compiles, runs, and produces the incorrect output. Incorrect output includes errors in the calculations, incorrect years of deposit, or errors in the amount deposited. Please note that calculating a deposit for every year instead of the “years to deposit” falls in this category.

-10 points There are data formatting errors.

no deductions The program compiles, runs, and generates the correct outputs (see the following page for a *minimal* example).

Additional notes:

Any loop structure is acceptable.

The enum **Family** must be used for input. AND it must have the correct properties as outlined above and below.

All financial calculations MUST use the **BigDecimal** data type.

```
Public enum Family

{

   WILMA(“Wilma”, “Flintstone”, 5000, 0.05, 10, 35),

   FRED(“Fred”, “Flintstone”, 15000, 0.075, 7, 30),

   BETTY(“Betty”, “Rubble”, 7500, 0.0375, 10, 25),

   BARNEY(“Barney”, “Rubble”, 5000, 0.09, 10, 35),


 . . .


     The properties’ “getters”/accessors should go here.


}
```

NB: WILMA deposits $5,000 a year at 5% interest for 10 years then keeps that money on account for a total of 35 years – likewise the data is in the same **order** for FRED, BETTY, & BARNEY even though the amounts are different.

More hints:

1. Take your time and build the enum first.
2. Make sure that your enum is accessible and the program can access the properties. (That is you could print out each FAMILY member’s specific data.)
3. Test your **BigDecimal** calculations before building the array that will hold the results.
4. IT IS PERFECTLY ACCEPTABLE to maintain the balance in the array, EVEN THOUGH no more interest is being accumulated. For example, BETTY & FRED don’t keep their deposit as long as WILMA & BARNEY. (hmmm…)

```
$java Assignment5
First name:          Wilma       Fred            Betty            Barney
Last name:           Flintstone  Flintstone      Rubble           Rubble
Annual Deposit:           5,000            7,500          15,000            5,000
Interest rate:               5%             7.5%           3.75%               9%
Years to deposit:            10                7              10               10
Total years:                 35               30              25               35
Year 1                    5,000            7,500          15,000            5,000
Year 2                    5,000            7,500          15,000            5,000
                  PLACEHOLDER DATA  - DETAILED OUTPUT TO FOLLOW
Year 35                   5,000            7,500          15,000            5,000
```