Sample Program 12.3

Using Objects

Problem statement. Modify Exercise 4.3's `mySavingsPlan1.py` by replacing its separate variables with a single object with data fields that replace the variables. Name the new file **`mySavingsPlanWithObjectsAndFunctions.py`**.

Create an object specification named **`class Savings`**, with the data fields for:

1. the amount deposited every month (D in previous versions)

2. interest rate (p in previous versions)

3. number of years to make deposits (years in previous versions)

4. the amount saved (S in previous versions)

Then in your program, declare an *object* of type **`Savings`***,* and use its data fields instead of the four separate variables. Include any additional data fields that you may wish to include beyond these four.

Also create a function named calculate, that takes a Savings object as its parameter, and performs all the calculations for the object. Call the function from main, instead of calculating in main.

Solution.

```
import math

class Savings:
  years = None # years of savings
  D = None # dollars deposited every month
  R = None # annual interest rate, percent
  p = None # monthly interest rate, decimal, calculated
  T = None # term of savings plan in months, calculated
  S = None # total saved with interest

def calculate(s):
  s.p = s.R / 100 / 12
  s.T = s.years * 12
  s.S = s.D * ((math.pow(1 + s.p, s.T) - 1) / s.p)


# create a savings plan object with initial values
savings = Savings()
savings.years = 10
savings.D =  100
savings.R = 7.5 # set years, D, and R only

# output (calculated) values
calculate(savings)

# echoing input values, unformatted
print("In", savings.years, "years, $", end = "")
print(savings.D, "deposited per month will grow to $", end = "")

# formatting output (see 4.1)
savings.S = "%.2f" % savings.S
print(savings.S, ".", sep = "")
```