**3.11** *(Modified* ***GradeBook*** *Class)* Modify class GradeBook (Fig. 3.10) as follows:

a) Include a String instance variable that represents the name of the course’s instructor.  
b) Provide a *set* method to change the instructor’s name and a *get* method to retrieve it.  
c) Modify the constructor to specify two parameters—one for the course name and one  
for the instructor’s name.  
d) Modify method displayMessage to output the welcome message and course name, followed by "This course is presented by: " and the instructor’s name.

Use your modified class in a test application that demonstrates the class’s new capabilities.

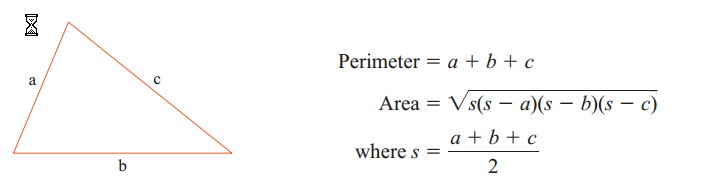
**3.12** *(Modified* ***Account*** *Class)* Modify class Account (Fig. 3.13) to provide a method called debit that withdraws money from an Account. Ensure that the debit amount does not exceed the Account’s balance. If it does, the balance should be left unchanged and the method should print a message indicating "Debit amount exceeded account balance." Modify class AccountTest (Fig. 3.14) to test the method debit.

**3.13** *(****Invoice*** *Class)* Create a class called Invoice that a hardware store might use to represent  
an invoice for an item sold at the store. An Invoice should include four pieces of information as  
instance variables—a part number (type String), a part description (type String), a quantity of the  
item being purchased (type int) and a price per item (double). Your class should have a constructor  
that initializes the four instance variables. Provide a *set* and a *get* method for each instance variable.  
In addition, provide a method named getInvoiceAmount that calculates the invoice amount (i.e.,  
multiplies the quantity by the price per item), then returns the amount as a double value.

If the  
quantity is not positive, it should be set to 0. If the price per item is not positive, it should be set to  
0.0. Write a test application named InvoiceTest that demonstrates class Invoice’s capabilities.

**3.15** *(****Date*** *Class)* Create a class called Date that includes three instance variables—a month (type  
int), a day (type int) and a year (type int). Provide a constructor that initializes the three instance  
variables and assumes that the values provided are correct. Provide a *set* and a *get* method for each instance variable. Provide a method displayDate that displays the month, day and year separated by forward slashes (/). Write a test application named DateTest that demonstrates class Date’s capabilities.

**Program** Define a class called Triangle that is capable of computing the perimeter and area of a triangle, given its three sides a, b, and c, as shown below. Notice that side b is the base of the triangle.



Also define a private method isValid to check the validity of three sides (The sides are valid, if the sum of any two sides is greater than the third side). If any one of them is invalid, the methods getArea and getPerimeter will return the message “INVALID\_DIMENSION”.

1. **NIFTY Sequence Generator:** Read a positive integer value, and compute the following sequence: If the number is even, halve it; if it's odd, multiply by 3 and add 1. Repeat this process until the value is 1, printing out each value. Finally print out how many of these operations you performed.

Typical output might be:

Inital value is 9

Next value is 28

Next value is 14

Next value is 7

Next value is 22

Next value is 11

Next value is 34

Next value is 17

Next value is 52

Next value is 26

Next value is 13

Next value is 40

Next value is 20

Next value is 10

Next value is 5

Next value is 16

Next value is 8

Next value is 4

Next value is 2

Final value 1, number of steps 19