UNIVERSITY OF BIRMINGHAM DUBAI

# LM Software Workshop 1 (34153, 34182, 34168, 36990)
## Lab Exercise Sheet
## Week 2 Chapter 3 (Loops & Selection Statements)

**The following 5 questions are mandatory for all students:**

1. Write a program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether the triangle is an equilateral triangle.

2. Write a program that accepts the lengths of three sides of a triangle as inputs. The program output should indicate whether the triangle is a right triangle. Recall from the Pythagorean theorem that in a right triangle, the square of one side equals the sum of the squares of the other two sides.

3. ~~Modify the guessing-game program of Section 3.5 so that the user thinks of a number that the computer must guess. The computer must make no more than the minimum number of guesses, and it must prevent the user from cheating by entering misleading hints. (Hint. Use the math.log function to compute the minimum number of guesses needed after the lower and upper bounds are entered.)~~

4. A standard science experiment is to drop a ball and see how high it bounces. Once the "bounciness" of the ball has been determined, the ratio gives a bounciness index. For example, if a ball dropped from a height of 10 feet bounces 6 feet high, the index is 0.6, and the total distance travelled by the ball is 16 feet after one bounce. If the ball were to continue bouncing, the distance after two bounces would be 10 ft + 6 ft + 6 ft + 3.6 ft = 25.6 ft. Note that the distance travelled for each successive bounce is the distance to the floor plus 0.6 of that distance as the ball comes back up. Write a program that lets the user enter the initial height from which the ball is dropped and the number of times the ball is allowed to continue bouncing. Output should be the total distance travelled by the ball.

5. A local biologist needs a program to predict population growth. The inputs would be the initial number of organisms, the rate of growth (a real number greater than 0), the number of hours it takes to achieve this rate, and several hours during which the population grows. For example, one might start with a population of 500 organisms, a growth rate of 2, and a growth period to achieve this rate of 6 hours. If none of the organisms die, this would imply that this population would double in size every 6 hours. Thus, after allowing 6 hours for growth, we would have 1000 organisms, and after 12 hours, we would have 2000 organisms. Write a program that takes these inputs and displays a prediction of the total population.

**(Challenge Questions) The following 2 questions are optional.**

6. The German mathematician Gottfried Leibniz developed the following method to approximate the value of :

$$\pi/4 = 1 - 1/3 + 1/5 - 1/7 + \ldots$$

Write a program that allows the user to specify the number of iterations used in this approximation and that displays the resulting value.

7. Teachers in most school districts are paid on a schedule that provides a salary based on their number of years of teaching experience. For example, a beginning teacher in the Lexington School District might be paid $30,000 the first year. For each year of experience after this first year, up to 10 years, the teacher receives a 2% increase over the preceding value. Write a program that displays a salary schedule, in tabular format, for teachers in a school district. The inputs are the starting salary, the percentage increase, and the number of years in the schedule. Each row in the schedule should contain the year number and the salary for that year.

**Complete the following case study (it's on page 73 of the textbook)**



## CASE STUDY: An Investment Report

It has been said that compound interest is the eighth wonder of the world. Our next case study, which computes an investment report, shows why.

### Request

Write a program that computes an investment report.

### Analysis

The inputs to this program are the following:

- An initial amount to be invested (a floating-point number)
- A period of years (an integer)
- An interest rate (a percentage expressed as an integer)

The program uses a simplified form of compound interest, in which the interest is computed once each year and added to the total amount invested. The output of the program is a report in tabular form that shows, for each year in the term of the investment, the year number, the initial balance in the account for that year, the interest earned for that year, and the ending balance for that year. The columns of the table are suitably labeled with a header in the first row. Following the output of the table, the program prints the total amount of the investment balance and the total amount of interest earned for the period. The proposed user interface is shown in Figure 3-1.

```
Enter the investment amount: 10000.00
Enter the number of years: 5
Enter the rate as a %: 5
Year  Starting balance  Interest  Ending balance
   1          10000.00    500.00        10500.00
   2          10500.00    525.00        11025.00
   3          11025.00    551.25        11576.25
   4          11576.25    578.81        12155.06
   5          12155.06    607.75        12762.82
Ending balance: $12762.82
Total interest earned: $2762.82
```

**Figure 3-1** The user interface for the investment report program

## Design

The four principal parts of the program perform the following tasks:

1. Receive the user's inputs and initialize data.
2. Display the table's header.
3. Compute the results for each year, and display them as a row in the table.
4. Display the totals.

The third part of the program, which computes and displays the results, is a loop. The following is a slightly simplified version of the pseudocode for the program, without the details related to formatting the outputs:

```
Input the starting balance, number of years, and interest rate
Set the total interest to 0.0
Print the table's heading
For each year
     compute the interest
     compute the ending balance
     print the year, starting balance, interest, and ending balance
     update the starting balance
     update the total interest
print the ending balance and the total interest
```

Note that `starting balance` refers to the original input balance and also to the balance that begins each year of the term. Ignoring the details of the output at this point allows us to focus on getting the computations correct. We can translate this pseudocode to a Python program to check our computations. A rough draft of a program is called a **prototype**. Once we are confident that the prototype is producing the correct numbers, we can return to the design and work out the details of formatting the outputs.

The format of the outputs is guided by the requirement that they be aligned nicely in columns. We use a format string to right-justify all of the numbers on each row of output. We also use a format string for the string labels in the table's header. After some trial and error, we come up with field widths of 4, 18, 10, and 16 for the year, starting balance, interest, and ending balance, respectively. We can also use these widths in the format string for the header.