Programmierübungen Gaddis-Buch

Chapter 2 Input, Processing, and Output

1. Personal Information

Write a program that displays the following information:

• Your name

• Your address, with city, state, and ZIP

• Your telephone number

• Your college major

2. Sales Prediction

A company has determined that its annual profit is typically 23 percent of total sales. Write a program that asks the user to enter the projected amount of total sales, then displays the profit that will be made from that amount.

Hint: Use the value 0.23 to represent 23 percent.

5. Distance Traveled

Assuming there are no accidents or delays, the distance that a car travels down the interstate can be calculated with the following formula:

Distance = Speed * Time

A car is traveling at 70 miles per hour. Write a program that displays the following:

• The distance the car will travel in 6 hours

• The distance the car will travel in 10 hours

• The distance the car will travel in 15 hours

8. Tip, Tax, and Total

Write a program that calculates the total amount of a meal purchased at a restaurant. The program should ask the user to enter the charge for the food, then calculate the amounts of a 18 percent tip and 7 percent sales tax. Display each of these amounts and the total.

11. Percentage of cats, lions and tigers

Write a program that reads the number of lions and tigers and calculates prints the percentage of lions and tigers.

ter1. Converting a decimal number into a binary number

Write a program that asks the user for an integer and converts this integer into a binary number.

Example: 10 -> 00001010

Hint: There exist different ways for implementation.

Chapter 3: Decision Structures and Boolean Logic

2. Areas of Rectangles

The area of a rectangle is the rectangle's length times its width. Write a program that asks for the length and width of two rectangles. The program should tell the user which rectangle has the greater area, or if the areas are the same.

6. Magic Dates

The date June 10, 1960, is special because when it is written in the following format, the month times the day equals the year:

6/10/60

Design a program that asks the user to enter a month (in numeric form), a day, and a two-digit year. The program should then determine whether the month times the day equals the year. If so, it should display a message saying the date is magic. Otherwise, it should display a message saying the date is not magic.

11. Book Club Points

Serendipity Booksellers has a book club that awards points to its customers based on the number of books purchased each month. The points are awarded as follows:

• If a customer purchases 0 books, he or she earns 0 points.
• If a customer purchases 2 books, he or she earns 5 points.
• If a customer purchases 4 books, he or she earns 15 points.
• If a customer purchases 6 books, he or she earns 30 points.
• If a customer purchases 8 or more books, he or she earns 60 points.

Write a program that asks the user to enter the number of books that he or she has purchased this month, then displays the number of points awarded.

12. Software Sales

A software company sells a package that retails for $99. Quantity discounts are given according to the following table:

Quantity	Discount
10–19	10%
20–49	20%
50–99	30%
100 or more	40%

Write a program that asks the user to enter the number of packages purchased. The

program should then display the amount of the discount (if any) and the total amount of the purchase after the discount.

15. February Days

The month of February normally has 28 days. But if it is a leap year (Schaltjahr), February has 29 days. Write a program that asks the user to enter a year. The program should then display the number of days in February that year. Use the following criteria to identify leap years:

1. Determine whether the year is divisible by 100. If it is, then it is a leap year if and only if it is also divisible by 400. For example, 2000 is a leap year, but 2100 is not.
2. If the year is not divisible by 100, then it is a leap year if and only if it is divisible by 4. For example, 2008 is a leap year, but 2009 is not.

Here is a sample run of the program:

Enter a year: 2008

In 2008 February has 29 days.

ter1. Rewriting grader.py

Rewrite `grader.py` (see chapter 3 programs) by using

- Version 1: the `elif` statement
- Version 2: the `match` / `case` statement (is a “switch”-like statement available from v3.10 on)

Chapter 4: Repetition Structures

2. Calories Burned

Running on a particular treadmill you burn 4.2 calories per minute. Write a program that uses a loop to display the number of calories burned after 10, 15, 20, 25, and 30 minutes.

3. Budget Analysis

Write a program that asks the user to enter the amount that he or she has budgeted for a month. A loop should then prompt the user to enter each of his or her expenses for the month and keep a running total. When the loop finishes, the program should display the amount that the user is over or under budget.

4. Distance Traveled

The distance a vehicle travels can be calculated as follows:

distance = *speed* x *time*

For example, if a train travels 40 miles per hour for three hours, the distance traveled is 120 miles. Write a program that asks the user for the speed of a vehicle (in miles per hour) and the number of hours it has traveled. It should then use a loop to display the distance the vehicle has traveled for each hour of that time period. Here is an example of the desired output:

What is the speed of the vehicle in mph? 40

How many hours has it traveled? 3

Hour	Distance Traveled
1	40
2	80
3	120

4. Table with Miles to Kilometers conversion

Create a table which creates kilometers for the respective miles. Consider the values 10, 20, 30, 40, 50, 60, 70, and 80.

8. Sum of Numbers

Write a program with a loop that asks the user to enter a series of positive numbers. The user should enter a negative number to signal the end of the series. After all the positive numbers have been entered; the program should display their sum.

ter1. Prime number

Write a program that reads an integer from the console and determines if the integer is a

prime number or not.

ter2. Prime number (using a Python package)

Write a program that uses a Python package for checking if an integer is a prime number or not. The Python Package Index (PyPI, https://pypi.org/) is a repository of software for the Python programming language.

ter3. Infinite for loop

Write a program with a for loop that never stops execution (infinite loop).

Chapter 5: Functions

1. Kilometer Converter

Write a program that asks the user to enter a distance in kilometers, then converts that distance to miles. The conversion formula is as follows:

Miles = Kilometers * 0.6214

4. Automobile Costs

Write a program that asks the user to enter the monthly costs for the following expenses incurred from operating his or her automobile: loan payment, insurance, gas, oil, tires, and maintenance. The program should then display the total monthly cost of these expenses, and the total annual cost of these expenses.

5. Property Tax

A county collects property taxes on the assessment value of property, which is 60 percent of the property's actual value. For example, if an acre of land is valued at $10,000, its assessment value is $6,000. The property tax is then 72¢ for each $100 of the assessment value. The tax for the acre assessed at $6,000 will be $43.20. Write a program that asks for the actual value of a piece of property and displays the assessment value and property tax.

9. Feet to Inches

One foot equals 12 inches. Write a function named feet_to_inches that accepts a number of feet as an argument and returns the number of inches in that many feet. Use the function in a program that prompts the user to enter a number of feet then displays the number of inches in that many feet.

11. Maximum of Two Values

Write a function named max that accepts two integer values as arguments and returns the value that is the greater of the two. For example, if 7 and 12 are passed as arguments to the function, the function should return 12. Use the function in a program that prompts the user to enter two integer values. The program should display the value that is the greater of the two.

17. Prime Numbers (see also the previous chapter)

A prime number is a number that is only evenly divisible by itself and 1. For example, the number 5 is prime because it can only be evenly divided by 1 and 5. The number 6, however, is not prime because it can be divided evenly by 1, 2, 3, and 6. Write a Boolean function named `is_prime` which takes an integer as an argument and returns true if the argument is

a prime number, or false otherwise. Use the function in a program that prompts the user to enter a number then displays a message indicating whether the number is prime.

ter1. Maximum of three numbers

Write a program that determines – using a function – the minimum of three integers which are read from the console.

ter2. Translating text

Write a program that translates single words, word fragments or full sentences from an arbitrary source language to English. The source language should be detected automatically. Texts to be translated into English should be entered via the console with entering “stop” for quitting the program. The translation itself should be coded using a function. For implementation install the package deep_translator (https://pypi.org/project/deep-translator/) and a solution which does not require an API Key, e.g. Google Translate. Assess the quality of the translations conducted.

Example:

```
C:\Users\raine\PycharmProjects\Python_Exercises_BA\.venv\Scripts\python.exe "C:\Users\raine\PycharmProjects\Python_Exercises_BA\Chapter_05\Exercise 5-ter2.py"
Please enter an arbitrary text for translation (stop to terminate): Je suis heureux d'apprendre Python!
I am happy to learn Python!
Please enter text: stop
Program terminated ...
```

Chapter 6: Introduction to File Input and Output

1. File Display

Assume a file containing a series of integers is named `numbers.txt` and exists on the computer's disk. Write a program that displays all of the numbers in the file.

2. File Head Display

Write a program that asks the user for the name of a file. The program should display only the first five lines of the file's contents. If the file contains less than five lines, it should display the file's entire contents.

3. Line Numbers

Write a program that asks the user for the name of a file. The program should display the contents of the file with each line preceded with a line number followed by a colon. The line numbering should start at 1.

5. Sum of Numbers

Assume a file containing a series of integers is named `numbers.txt` and exists on the computer's disk. Write a program that reads all of the numbers stored in the file and calculates their total.

7. Random Number File Writer

Write a program that writes a series of random numbers to a file. Each random number should be in the range of 1 through 500. The application should let the user specify how many random numbers the file will hold.

Chapter 7: Lists and Tuples

2. Lottery Number Generator

Design a program that generates a seven-digit lottery number. The program should generate seven random numbers, each in the range of 0 through 9, and assign each number to a list element. (Random numbers were discussed in Chapter 5.) Then write another loop that displays the contents of the list.

4. Number Analysis Program

Design a program that asks the user to enter a series of 20 numbers. The program should store the numbers in a list then display the following data:

- The lowest number in the list
- The highest number in the list
- The total of the numbers in the list
- The average of the numbers in the list

5. Charge Account Validation

Create a text file `named charge_accounts.txt`. This file contains (created by offline hand) a list of a company's valid charge account numbers. Each account number is a seven-digit number, such as 5658845.

Write a program that reads the contents of the file into a list. The program should then ask the user to enter a charge account number. The program should determine whether the number is valid by searching for it in the list. If the number is in the list, the program should display a message indicating the number is valid. If the number is not in the list, the program should display a message indicating the number is invalid.

6. Rolling the dice

Create a program which asks the use how many times the dice should be rolled. The roll function accepts this number and returns a sorted list of num random numbers between 1 and 6.

Chapter 8: More About Strings

1. Initials

Write a program that gets a string containing a person's first, middle, and last names, and displays their first, middle, and last initials. For example, if the user enters John William Smith, the program should display J. W. S.

2. Sum of Digits in a String

Write a program that asks the user to enter a series of single-digit numbers with nothing separating them. The program should display the sum of all the single digit numbers in the string. For example, if the user enters 2514, the method should return 12, which is the sum of 2, 5, 1, and 4.

6. Average Number of Words

If you have downloaded the source code from the Computer Science Portal you will find a file named `text.txt` in the Chapter 08 folder. The text that is in the file is stored as one sentence per line. Write a program that reads the file's contents and calculates the average number of words per sentence.

7. File statistics

Read again `text.txt` and determine for each character in the file if it is uppercase, lowercase, a digit, or space. Print a running total of each.

10. Most Frequent Character

Write a program that lets the user enter a string and displays the character that appears most frequently in the string.

Chapter 9: Dictionaries and Sets

2. Capital Quiz

Write a program that creates a dictionary containing states as keys, and their capitals as values. (Use the Internet to get a list of the states and their capitals.) The program should then randomly quiz the user by displaying the name of a state and asking the user to enter that state's capital. The program should keep a count of the number of correct and incorrect responses.

4. Unique Words

Write a program that opens a specified text file then displays a list of all the unique words found in the file.

Hint: Store each word as an element of a set.

5. Number Frequency

Write a program that generates 100 random integers between 1 and 10. Store each number in a dictionary where the number is the key and the frequency of occurrence is the value. After all numbers are generated, print a table with the header "Number Frequency". Sort the dictionary by the numbers in ascending order and display each number with its frequency in tab-separated format.