

Answers to Questions and Exercises

This document contains answers to in-text questions and exercises in the book ***Learning to Program in Python*** by PM Heathcote.

The Python programs referred to in the text and in this Answers document are all in a separate folder within this pack.

Copyright

© 2017 PG Online Limited

The contents of this pack are protected by copyright. No part of the materials distributed with this unit may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic or otherwise, without the prior written permission of PG Online Limited.

Answers

Chapter 1 – Data types, operators and I-O

# In-text questions

**Q1:** (a) 10\*6 + 5\*3 will give the result directly in interactive mode

or use a print statement:

>>> print(10\*6 + 5\*3)

75

(b) 10\*(6 + 5)\*3 will give the result directly in interactive mode

or use a print statement:

>>> print(10\*(6+5)\*3)

330

(c) >>> 33%7

5

**Q2:** >>> result = round(6.7/3, 3)

>>> print (result)

2.233

*(or any variation that produces the same result)*

**Q3:** (a) highScore = 25

(b) playerName = “Davina”

(c) height = 4.5

base = 6

print (base\*height/2)

**Q4:** (a) counter += 1

(b) housePrice \*= 2

(c) hits -= penalty

(d) totalCostOfMeal /= 3

**Q5:** (a) print (“More pain, more gain”)

(b) print (“That’s true!)

(c) print (“Pelham” + “123”)

(d) How will this statement be printed?

# Exercises

1. (a) >>> food = input("Enter favourite food: ")

Enter favourite food: peanut butter

>>> colour = input("Enter favourite colour: ")

Enter favourite colour: blue

>>> print ("My favourite food is " + food + " and my favourite colour is "

+ colour)

My favourite food is peanut butter and my favourite colour is blue

(b) >>> print ("My favourite food is", food, "and my favourite colour is",

colour)

My favourite food is peanut butter and my favourite colour is blue

2. (a) Please enter your name: Jo

>>> telephone = input ("Please enter your telephone number: ")

Please enter your telephone number: 01227 342567

>>> print(name + "\n\n" + telephone)

Jo

01227 342567

(b) >>> print(name, "\t", telephone)

3. (a) >>> 40 // 11

3

(b) >>> 40 % 11

7

(c) >>> 2\*\*10

1024

(d) >>> "three" > "two"

False

(e) >>> "abc" < "ABC"

False

(f) >>> 1<=4 and 7<=7

True

(g) >>> "Fred" != "fred"

True

Chapter 2 – Strings and numbers

# In-text questions

**Q1**: method upper() should be followed by parentheses

print ( "\nUppercase string:", astring.upper())

next line missing a closing bracket

print("\nNew string:", astring.replace("good","excellent"))

# Exercises

1. See Python program Ch 2 Exercise 1 string methods.py

2. See Python program Ch 2 Exercise 2 radius and circumference.py

3. See Python program Ch 2 Exercise 3 rounding floating point numbers.py

Chapter 3 – Selection

# In-text questions

**Q1:** See Python program Ch 3 Question 1 minimum number.py

num1 = int(input("Enter first number: "))

num2 = int(input("Enter second number: "))

if num1 < num2:

minimum = num1

else:

minimum = num2

print("minimum number =",minimum)

**Q2:** weekendRate will remain False.

See Python program Ch 3 Example 3 Selection statements.py

# Exercises

1. See Python program Ch 3 Exercise 1 area of square or rectangle.py

2. See Python program Ch 3 Exercise 2 menu of subject choices.py

3. See Python program Ch 3 Exercise 3 throw two dice.py

4. See Python program Ch 3 Exercise 4 discount on goods purchased.py

5. See Python program Ch 3 Exercise 5 car parking charges.py

Chapter 4 – Iteration

# In-text questions

**Q1:** (a) See Python program Ch 4 Question 1a Lift-off.py

(b) See Python program Ch 4 Question 1b total and average.py Q2

**Q2:** The maximum result would be correct so long as all the results are 100 or less. The minimum result would be given as -1.

See Python program Ch 4 Question 2 max and min error.py

**Q3:** Maximum test result 56

Minimum test result 100

“100” is less than “19” when these are interpreted as strings.

See Python program Ch 4 Question 3 max and min with string input.py

**Q4:** See Python program Ch 4 Question 4 max and min no results entered.py

**Q5:** See Python program Ch 4 Question 5 find e.py

# Exercises

1. See Python program Ch 4 Exercise 1 10 random numbers.py

2. See Python program Ch 4 Exercise 2 1000 sets of 10 random numbers.py

The answer should be close to 55. Overall, each of the numbers 1-10 should be generated with the same frequency. The total of the numbers 1 to 10 is 55, so the average total of all the numbers will be close to this. Try it with 10,000 sets of numbers – it should be even closer. Obtaining an answer to a question like “What is the total of the numbers 1-10?” using this method is called the Monte Carlo method, and is a useful technique to find answers to questions for which there is no obvious formula.

3. See Python program Ch 4 Exercise 3 product codes.py

Chapter 5 – Lists and tuples

# In-text questions

**Q1:** The program will crash with an “index out of range” error

**Q2:** list3 = list1 + list2

**Q3:** last = list1.pop()

**Q4:** list4 = list3[3:7]

**Q5:** monthlyAvg[10][2]

**Q6:** print(monthlyAvg[11])

**Q7:** (a) 12

(b) 3

# Exercises

1. (a) alist =[]

(b) alist.append(“apple”)

(c) blist = [0] \* 20

(d) clist = alist + blist

(e) lastC = clist.pop()

See Python program Ch 5 Exercise 1 list operations.py

2. See Python program Ch 5 Exercise 2 list slice.py

3. See Python program Ch 5 Exercise 3 student marks.py

4. See Python program Ch 5 Exercise 4 2-D temperatures.py

5. See Python program Ch 5 Exercise 5 top game score.py

Chapter 6 – Validating user input

# In-text questions

**Q1:** prodCode = input(“Please enter 6-character product code: ”)

while len(prodCode) != 6:

prodCode = input(“Invalid code – please re-enter: “:”)

# Exercises

1. See Python program Ch 6 Exercise 1 theatre tickets.py

2. See Python program Ch 6 Exercise 2 validate car reg.py

3. See Python program Ch 6 Exercise 3 verify email address .py

4. See Python program Ch 6 Exercise 4 validate password.py

and Ch 6 Exercise 4 validate password v2.py

5. See Python program Ch 6 Exercise 5 Caesar cipher.py

Chapter 7 – Searching and sorting

# In-text questions

**Q1:** print(studentMarks[“Robina”])

**Q2:** studentMarks[2][0]

**Q3:** studentNamesAscending = sorted(studentMarks, key=lambda x:x[0])

See Python program Ch 7 Question 3 sort 2-D list.py

# Exercises

1. See Python program Ch 7 Exercise 1 anagrams.py

2. See Python program Ch 7 Exercise 2 dictionary of telephone numbers.py

3. See Python program Ch 7 Exercise 3 temperatures in 2-D list.py

4. See Python program Ch 7 Exercise 4 processing 2-D list.py

Chapter 8 – Functions

# In-text questions

**Q1:** age = input(“Input your age: ”)

**Q2:** Hello Jo *(or whatever name was entered by the user)*

Jo is Player 1

**Q3:** def birthday(name, age):

print("Happy birthday, " + name + "! You're " + str(age) + " today!")

**Q4:** The last line printed will be

In main program, name = Bob

# Exercises

1. See Python program Ch 8 Exercise 1 cube function.py

2. See Python program Ch 8 Exercise 2 encrypt a message.py

3. See Python program Ch 8 Exercise 3 max temperature.py

4. See Python program Ch 8 Exercise 4 Weekly pay.py

Chapter 9 – Reading and writing files

# In-text questions

**Q1:** (a) See Python program Ch 9 Question 1 read and print temperature file.py

(b) London,7,1200,

Accra,30,1200,

Baghdad,20,1500,

Winnipeg,-12,0600,

New York,14,0700,

Nairobi,27,1500,

Sydney,22,2300,

**Q2:** See Python program Ch 9 Question 2 integer temperatures.py

**Q3:** a = 3

b = 4

c = a + b

d = a \* b

print("%d + %d = %d" %(a,b,c))

print("The product of %d and %d is %d" %(a,b,d))

See Python program Ch 9 format operators.py

# Exercises

1. See Python program Ch 9 Exercise 1 create or append to student file.py

2. See Python program Ch 9 Exercise 2 girls in Yr 10.py

3. See Python program Ch 9 Exercise 3 student records in order of surname.py

4. (No answer – this is a “discovery” exercise.)

Chapter 10 – Databases and SQL

# Exercises

1. CREATE TABLE tblScores

(

playerID TEXT,

firstname TEXT,

surname TEXT,

score1 INTEGER,

score2 INTEGER,

score3 INTEGER,

primary key (playerID)

)

(Lowercase *text*, *integer* is also acceptable)

2. INSERT INTO tblScores(playerID, firstname, surname, score1, score2, score3)

VALUES (“MF123”, “Maria”, “Ferdinand”,0, 0, 0)

3. UPDATE tblScores

SET score1 = 87, score2 = 79, score3 = 63

WHERE playerID = “MF123”

4. SELECT score1

FROM tblScores

where playerID = “MF123”

5. SELECT \*

FROM tblScores

where playerID = “MF123”

Chapter 11 – Database programming with SQLite

# Exercises

1. See Python program Ch 11 Exercise 1 print tblFilms.py

2. See Python program Ch 11 Exercise 2 film database maintenance.py

3. See Python program Ch 11 Exercise 3 print film database reports.py

Chapter 12 – Introduction to TKInter

# Exercises

1. See Python program Ch 12 Exercise 1 Hi there.py

2. (a) See Python program 12 Exercise 2a left and right labels.py

(b) See Python program Ch 12 Exercise 2b left and right justify.py

Chapter 13 – Developing an application using TKInter

# Exercises

1. See Python programs Ch 13 Exercise 1 student details to database.py

and Ch 13 Exercise 1 student details to file.py

2. See Python program Ch 13 Exercise 2 Print student details.py

3. See Python program Ch 13 Exercise 3 password validation.py

4. See Python program Ch 13 Exercise 4 Write test to file.py

Chapter 14 – Program design

# Exercises

1. See Python program Ch 14 Exercise 1 calculate area.py

2. See Python program Ch 14 Exercise 2 multiple choice test options.py

Chapter 15 – Testing and debugging

# In-text questions

**Q1:** for performance in range(5): Should be range(4)

if number in range (120): should be range(121)

The if flag: conditional statement is incorrect.

(Compare with Ch 6 Exercise 1 Theatre tickets)

# Exercises

1. As it stands, the program adds 1 to the variable passwordChecks each time a valid lowercase or uppercase letter is detected. It also adds 1 to passwordChecks if the password length is within the acceptable range. However, it will accept any password with at least 3 valid upper- or lowercase letters.

There needs to be separate flags for each condition, i.e. between 8 and 15 letters, an uppercase letter, a lowercase letter, and a number.

See Python program Ch 6 Exercise 4 validate password.py

2. The program works correctly if the correct password is entered on the first attempt.

If the user enters an incorrect password, the statement

input("Password incorrect - please re-enter: ")

is executed. However even when the correct password is entered, it is not accepted.

The input should be assigned to the variable password, i.e.

password =input("Password incorrect - please re-enter: ")

Note that there is no limit to the number of tries the user can have in this version.

See Python program Ch 6 Exercise 5 Caesar cipher.py

Every effort has been made to ensure that all answers are correct. Any errors should be reported directly to support@pgonline.co.uk and changes will be made in any subsequent editions of the material.

Artwork



Fosse No. 4 © 2015 Barbara Burns

Oil on linen, 30x30cm

www.slaneart.com