# Introduction to Programming Solutions

# 9

## Solution to Exercise 1: Mailing Address

```
##
# Display a person's complete mailing address.
#
print("Ben Stephenson")
print("Department of Computer Science")
print("University of Calgary")
print("2500 University Drive NW")
print("Calgary, Alberta T2N 1N4")
print("Canada")
```

## Solution to Exercise 3: Area of a Room

```
##
# Compute the area of a room.
#

# Read the input values from the user
length = float(input("Enter the length of the room in feet: "))
width = float(input("Enter the width of the room in feet: "))

# Compute the area of the room
area = length * width

# Display the result
print("The area of the room is", area, "square feet")
```

The `float` function is used to convert the user's input into a number.

In Python, multiplication is performed using the * operator.



B. Stephenson, *The Python Workbook*, DOI 10.1007/978-3-319-14240-1_9

## Solution to Exercise 4: Area of a Field

```
##
#  Compute the area of a field, reporting the result in acres.
#
SQFT_PER_ACRE = 43560

# Read input from the user
length = float(input("Enter the length of the field in feet: "))
width = float(input("Enter the width of the field in feet: "))

# Compute the area in acres
acres = length * width / SQFT_PER_ACRE

# Display the result
print("The area of the field is", acres, "acres")
```

## Solution to Exercise 5: Bottle Deposits

```
##
#  Compute the refund amount for a collection of bottles.
#
LESS_DEPOSIT = 0.10
MORE_DEPOSIT = 0.25

# Read input from the user
less = int(input("How many containers 1 litre or less do you have? "))
more = int(input("How many containers more than 1 litre do you have? "))

# Compute the refund amount
refund = less * LESS_DEPOSIT + more * MORE_DEPOSIT

# Display the result
print("Your total refund will be $%.2f." % refund)
```

The `%.2f` format specifier indicates that a value should be formatted as a floating point number with 2 digits to the right of the decimal point.

## Solution to Exercise 6: Tax and Tip

```
##
#  Compute the tax and tip for a restaurant meal.
#
TAX_RATE = 0.05
TIP_RATE = 0.18

# Read the cost of the meal from the user
cost = float(input("Enter the cost of the meal: "))

# Compute the tax and the tip
tax = cost * TAX_RATE
tip = cost * TIP_RATE
total = cost + tax + tip

# Display the result
print("The tax is %.2f and the tip is %.2f, making the total %.2f" % \
      (tax, tip, total))
```

My local tax rate is 5%. In Python we represent 5% and 18% as 0.05 and 0.18 respectively.

The \ at the end of the line is called the line continuation character. It tells Python that the statement continues on the next line. Do not include any spaces or tabs after the \ character.

## Solution to Exercise 7: Sum of the First *n* Positive Integers

```
##
#  Compute the sum of the first n positive integers.
#

# Read input from the user
n = int(input("Enter a positive integer: "))

# Compute the sum
sm = n * (n + 1) / 2

# Display the result
print("The sum of the first", n, "positive integers is", sm)
```

Python includes a built-in function named `sum`. As a result, we will use a different name for our variable.

## Solution to Exercise 10: Arithmetic

```
##
#  Demonstrate Python's mathematical operators and its math module.
#
from math import log10

# Read input values from the user
a = int(input("Enter the value of a: "))
b = int(input("Enter the value of b: "))

# Compute and display the sum, difference, product,
# quotient and remainder
print(a, "+", b, "is", a + b)
print(a, "-", b, "is", a - b)
print(a, "*", b, "is", a * b)
print(a, "/", b, "is", a / b)
print(a, "%", b, "is", a % b)

# Compute the logarithm and the power
print("The base 10 logarithm of", a, "is", log10(a))
print(a, "^", b, "is", a**b)
```

We must import the `log10` function from the `math` module before we call it. Import statements normally appear at the top of the file.

The remainder is computed using the `%` operator.

## Solution to Exercise 13: Making Change

```
##
#  Compute the minimum collection of coins needed to represent a number of cents.
#
CENTS_PER_TOONIE = 200
CENTS_PER_LOONIE = 100
CENTS_PER_QUARTER = 25
CENTS_PER_DIME =    10
CENTS_PER_NICKEL =   5

# Read the number of cents from the user
cents = int(input("Enter the number of cents: "))
```

```
# Determine the number of toonies by performing an integer division by 200.  Then compute
# the amount of change that still needs to be considered by
# computing the remainder after dividing by 200.
print(" ", cents // CENTS_PER_TOONIE, "toonies")
cents = cents % CENTS_PER_TOONIE

# Repeat the process for loonies, quarters, dimes, and nickels
print(" ", cents // CENTS_PER_LOONIE, "loonies")
cents = cents % CENTS_PER_LOONIE

print(" ", cents // CENTS_PER_QUARTER, "quarters")
cents = cents % CENTS_PER_QUARTER

print(" ", cents // CENTS_PER_DIME, "dimes")
cents = cents % CENTS_PER_DIME

print(" ", cents // CENTS_PER_NICKEL, "nickels")
cents = cents % CENTS_PER_NICKEL

# Display the number of pennies
print(" ", cents, "pennies")
```

Integer division, which discards any fractional part of the result, is performed using the // operator.

## Solution to Exercise 14: Height Units

```
##
#  Convert a height in feet and inches to centimeters.
#
IN_PER_FT = 12
CM_PER_IN = 2.54

# Read input from the user
print("Enter your height:")
feet = int(input("  Number of feet: "))
inches = int(input("  Number of inches: "))

# Compute the equivalent number of centimeters
cm = (feet * IN_PER_FT + inches) * CM_PER_IN

# Display the result
print("Your height in centimeters is:", cm)
```

## Solution to Exercise 17: Heat Capacity

```
##
#  Compute the amount of energy needed to heat a volume of water, and the cost of doing so.
#

# Define constants for the specific heat capacity of water and the price of electricity
WATER_HEAT_CAPACITY = 4.186
ELECTRICITY_PRICE = 8.9
J_TO_KWH = 2.777e-7
```

> Python allows numbers to be written in scientific notation by placing the coefficient to the left of an `e` and the exponent to its right. As a result, $2.777 * 10^{-7}$ is written as `2.777e-7`.

```
# Read the volume from the user
volume = float(input("Enter the amount of water in milliliters: "))
d_temp = float(input("Enter the temperature increase (degrees Celsius): "))

# Compute the energy in Joules
q = volume * d_temp * WATER_HEAT_CAPACITY

# Display the result in Joules
print("That will require %d Joules of energy." % q)

# Compute the cost
kwh = q * J_TO_KWH
cost = kwh * ELECTRICITY_PRICE

# Display the cost
print("That much energy will cost %.2f cents." % \
      cost)
```

> Because water has a density of 1 gram per milliliter grams and milliliters can be used interchangeably. Prompting the user for milliliters makes the program easier to use because most people think about the volume of water in a coffee cup, not its mass.

## Solution to Exercise 19: Free Fall

```
##
#  Compute the speed of an object when it hits the ground after being dropped.
#
from math import sqrt

# Define a constant for the acceleration due to gravity in m/s**2
GRAVITY = 9.8

# Read the height from which the object is dropped
d = float(input("Height from which the object is dropped (in meters): "))
```

```
# Compute the final velocity
vf = sqrt(2 * GRAVITY * d)

# Display the result
print("It will hit the ground at %.2f m/s." % vf)
```

The $v_i^2$ term has not been included in the calculation because $v_i$ is 0.

## Solution to Exercise 23: Area of a Regular Polygon

```
##
#  Compute the area of a regular polygon.
#
from math import tan, pi

# Read input from the user
s = float(input("Enter the length of each side of the polygon: "))
n = int(input("Enter the number of sides: "))

# Compute the area of the polygon
area = (n * s ** 2) / (4 * tan(pi / n))

# Display the result
print("The area of the polygon is", area)
```

We have chosen to cast input `n` to an integer because a polygon cannot have a fractional number of sides.

## Solution to Exercise 25: Units of Time (Again)

```
##
#  Convert a number of seconds to days, hours, minutes and seconds.
#
SECONDS_PER_DAY = 86400
SECONDS_PER_HOUR = 3600
SECONDS_PER_MINUTE = 60

# Read input from the user
seconds = int(input("Enter a number of seconds: "))

# Compute the days, hours, minutes and seconds
days = seconds / SECONDS_PER_DAY
seconds = seconds % SECONDS_PER_DAY

hours = seconds / SECONDS_PER_HOUR
seconds = seconds % SECONDS_PER_HOUR

minutes = seconds / SECONDS_PER_MINUTE
seconds = seconds % SECONDS_PER_MINUTE

# Display the result with the desired formatting
print("The equivalent duration is", \
      "%d:%02d:%02d:%02d." % (days, hours, minutes, seconds))
```

The `%02d` format specifier tells Python to format the integer using two digits, adding a leading 0 if necessary.

## Solution to Exercise 28: Wind Chill

```
##
# Compute the wind chill index for a given air temperature and wind speed
#
WC_OFFSET = 13.12
WC_FACTOR1 = 0.6215
WC_FACTOR2 = -11.37
WC_FACTOR3 = 0.3965
WC_EXPONENT = 0.16
```

Computing wind chill requires several numeric constants that were determined by scientists and medical experts.

```
# Read the air temperature and wind speed from the user
temp = float(input("Enter the air temperature (degrees Celsius): "))
speed = float(input("Enter the wind speed (kilometers per hour): "))

# Compute the wind chill index
wci = WC_OFFSET + \
      WC_FACTOR1 * temp + \
      WC_FACTOR2 * speed ** WC_EXPONENT + \
      WC_FACTOR3 * temp * speed ** WC_EXPONENT

# Display the result rounded to the closest integer
print("The wind chill index is", round(wci))
```

## Solution to Exercise 32: Sort 3 Integers

```
##
# Sort 3 values entered by the user into increasing order.
#

# Read the numbers from the user, naming them a, b and c
a = int(input("Enter the first number: "))
b = int(input("Enter the second number: "))
c = int(input("Enter the third number: "))
```

```
mn = min(a,b,c)              # the minimum value
mx = max(a,b,c)              # the maximum value
md = a + b + c - mn - mx     # the middle value

# Display the result
print("The numbers in sorted order are:")
print(" ", mn)
print(" ", md)
print(" ", mx)
```

Since `min` and `max` are the names of functions in Python we shouldn't use those names for variables. Instead we use variables named `mn` and `mx` to hold the minimum and maximum values respectively.

## Solution to Exercise 33: Day Old Bread

```
##
# Compute the price of a day old bread order.
#
BREAD_PRICE = 3.49
DISCOUNT_RATE = 0.60

# Read the number of loaves from the user
num_loaves = int(input("Enter the number of day old loaves: "))

# Compute the discount and total price
regular_price = num_loaves * BREAD_PRICE
discount = regular_price * DISCOUNT_RATE
total = regular_price - discount

# Display the result
print("Regular price: %5.2f" % regular_price)
print("Discount:      %5.2f" % discount)
print("Total:         %5.2f" % total)
```

The `%5.2f` format tells Python that a total of at least 5 spaces should be used to display the number, with 2 digits to the right of the decimal point. This will help keep the columns lined up when the number of digits needed for the discount and the total are different.