Programming Exercises

The exercises in this section are optional and do not report to the performance dashboard. Instructors can decide whether to assign these exercises and students can check the correctness of their programs using the Check Exercise tool.

Section 4.2

4.1 (*Geometry: area of a pentagon*) Write a program that prompts the user to enter the length from the center of a pentagon to a vertex and computes the area of the pentagon, as shown in the following figure.

The formula for computing the area of a pentagon is $Area = \frac{5 \times s^2}{4 \times \tan\left(\frac{\pi}{5}\right)}$, where s is the length of a side. The side can be computed using the formula $s = 2r \sin \frac{\pi}{5}$, where r is the length from the center of a pentagon to a vertex. Round up two digits after the decimal point.

***4.2** (*Geometry: great circle distance)* The great circle distance is the distance between two points on the surface of a sphere. Let $(x1, y1)$ and $(x2, y2)$ be the geographical latitude and longitude of two points. The great circle distance between the two points can be computed using the following formula:

$$d = radius \times \arccos(\sin(x_1) \times \sin(x_2) + \cos(x_1) \times \cos(x_2) \times \cos(y_1 - y_2))$$

Write a program that prompts the user to enter the latitude and longitude of two points on the earth in degrees and displays its great circle distance. The average earth radius is 6,371.01 km. Note that you need to convert the degrees into radians using the `math.radians` function since the Python trigonometric functions use radians. The

latitude and longitude degrees in the formula are for north and west. Use negative to indicate south and east degrees.

***4.3** (*Geography: estimate areas*) Use the GPS locations for Atlanta, Georgia; Orlando, Florida; Savannah, Georgia; and Charlotte, North Carolina in the figure in Section 4.1 to compute the estimated area enclosed by these four cities. (Hint: Use the formula in Programming Exercise 4.2 to compute the distance between two cities. Divide the polygon into two triangles and use the formula in Programming Exercise 2.14 to compute the area of a triangle.)

4.4 (*Geometry: area of a hexagon*) The area of a *hexagon* can be computed using the following formula (`s` is the length of a side):

$$Area = \frac{6 \times s^2}{4 \times \tan\left(\frac{\pi}{6}\right)}$$

Write a program that prompts the user to enter the side of a *hexagon* and displays its area.

***4.5** (*Geometry: area of a regular polygon*) A regular polygon is an n-sided polygon in which all sides are of the same length and all angles have the same degree (i.e., the polygon is both equilateral and equiangular). The formula for computing the area of a regular polygon is

$$Area = \frac{n \times s^2}{4 \times \tan\left(\frac{\pi}{n}\right)}$$

Here, `s` is the length of a side. Write a program that prompts the user to enter the number of sides and their length of a regular polygon and displays its area.

Sections 4.7–4.8

**4.6 (*Turtle: draw a star*) Write a program that prompts the user to enter the length of the star and draw a star, as shown in Figure 4.8a. (Hint: The inner angle of each point in the star is 36 degrees.)

Figure 4.8

The program (a) draws a star, (b) displays a STOP sign, and (c) draws an Olympic symbol.

(Screenshots courtesy of Apple.)

*4.7 (*Turtle: display a STOP sign*) Write a program that displays a STOP sign, as shown in Figure 4.8b. The hexagon is in red and the text is in white.

4.8 (*Turtle: draw the Olympic symbol*) Write a program that prompts the user to enter the radius of the rings and draws an Olympic symbol of five rings of the same size with the colors blue, black, red, yellow, and green, as shown in Figure 4.8c.

*4.9 (*Turtle: paint a smiley face*) Write a program that paints a smiley face, as shown in Figure 4.9a.

Figure 4.9

The program paints a smiley face in (a) and draws five shapes with bottom edges parallel to the x-axis in (b).

(Screenshots courtesy of Apple.)

**4.10 (*Turtle: draw shapes*) Write a program that draws a triangle, square, pentagon, hexagon, and octagon, as shown in Figure 4.9b. Note that the bottom edges of these shapes are parallel to the *x*-axis. (Hint: For a triangle with a bottom line parallel to the *x*-axis, set the turtle's heading to 60 degrees.)

**4.11 (*Turtle: triangle area*) Write a program that prompts the user to enter the three points p1, p2, and p3 for a triangle and display its area below the triangle, as shown in Figure 4.10a. The formula for computing the area of a triangle is given in Programming Exercise 2.14.

Figure 4.10

The program displays (a) the area of the triangle, (b) draws a line, and (c) displays the quadrilateral angles.

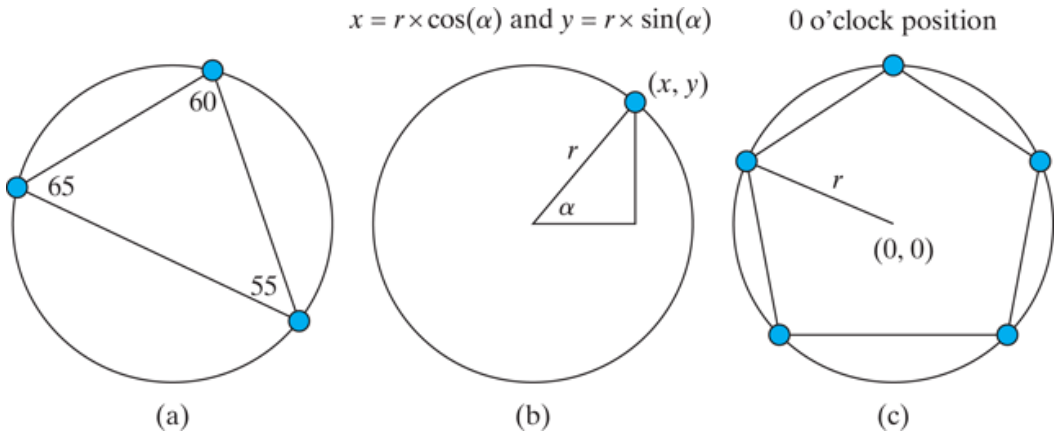
(Screenshots courtesy of Apple.)

4.12 (*Turtle: draw a line*) Write a program that prompts the user to enter two points and draw a line to connect the points and displays the coordinates of the points, as shown in Figure 4.10b.

***4.13** (*Turtle: quadrilateral angles*) A quadrilateral is a polygon with four sides. Write a program that prompts the user to enter the four points p1, p2, p3, and p4 in a counter-clockwise order and display its angles, as shown in Figure 4.10c. Hint: Divide the quadrilateral into two triangle and use the formula in Listing 4.2 to compute the angles.

***4.14** (*Random point on a circle*) Write a program that generates three random points on a circle centered at (0, 0) with radius 40 and display three angles in a triangle formed by these three points, as shown in Figure 4.11a. (Hint: Generate a random angle α in radians between 0 and 2π, as shown in Figure 4.11b and the point determined by this angle is $(r * \cos(\alpha), r * \sin(\alpha))$.)

Figure 4.11



(a) A triangle is formed from three random points on the circle. (b) A random point on the circle can be generated using a random angle α. (c) A pentagon is centered at (0, 0) with one point at the 0 o'clock position.

***4.15** (*Corner point coordinates*) Suppose a pentagon is centered at (0, 0) with one point at the 0 o'clock position, as shown in Figure 4.11c. Write a program that prompts the user to enter the radius of the bounding rectangle of a pentagon and displays the coordinates of the five corner points on the pentagon.

***4.16** (*Turtle: random point on a circle*) Revise Programming Exercise 4.14 to display three random points on the circle, as shown in Figure 4.12a. These three points form a

triangle and display the triangle.

Figure 4.12

(a) A triangle is displayed with three random points on the circle. (b) A pentagon displayed is centered at (0, 0) with one point at the 0 o'clock position.

(Screenshots courtesy of Apple.)

***4.17** (*Turtle: display a pentagon*) Revise Programming Exercise 4.15 to prompt the user to enter the radius of the bounding circle of a pentagon and display the pentagon, as shown in Figure 4.12b.

Sections 4.3–4.6

***4.18** (*Find the character of an ASCII code*) Write a program that receives an ASCII code (an integer between `0` and `127`) and displays its character. For example, if the user enters `97`, the program displays the character `a`.

***4.19** (*Find the ASCII code of a character*) Write a program that receives a character and displays its ASCII code.

4.20 (*Random character*) Write a program that displays a random uppercase letter.

***4.21** (*Financial application: payroll*) Write a program that reads the following information and prints a payroll statement:

Employee's name (e.g., Smith)
Number of hours worked in a week (e.g., 10)
Hourly pay rate (e.g., 9.75)
Federal tax withholding rate (e.g., 20%)
State tax withholding rate (e.g., 9%)

***4.22** (*Turtle: display Unicodes*) Write a program to display Greek letters **αβγδεζηθ**. The Unicode of these characters are `\u03b1 \u03b2 \u03b3 \u03b4 \u03b5 \u03b6 \u03b7 \u03b8`.

***4.23** (*Convert letter grade to number*) Write a program that prompts the user to enter a letter grade A/a, B/b, C/c, D/d, or F/f and displays its corresponding numeric value 4, 3, 2, 1, or 0.

***4.24** (*Vowel or consonant?*) Assume letters `A/a`, `E/e`, `I/i`, `O/o`, and `U/u` as the vowels. Write a program that prompts the user to enter a letter and check whether the letter is a vowel or consonant.

***4.25** (*Days of a month*) Write a program that prompts the user to enter the year and the first three letters of a month name (with the first letter in uppercase) and displays the number of days in the month.

***4.26** (*Student major and status*) Write a program that prompts the user to enter two characters and displays the major and status represented in the characters. The first character indicates the major and the second is number character 1, 2, 3, and 4, which indicates whether a student is a freshman, sophomore, junior, or senior. Suppose the following characters are used to denote the majors:

M: Mathematics

C: Computer Science

I: Information Technology

***4.27** (*Phone key pads)* The international standard letter/number mapping found on the telephone is shown below:

Write a program that prompts the user to enter a lowercase or uppercase letter and displays its corresponding number.

4.28 (*Process a string*) Write a program that prompts the user to enter a string and displays its length and its first character.

4.29 (*Business: check ISBN-10*) An **ISBN-10** (International Standard Book Number) consists of 10 digits:
$d_1d_2d_3d_4d_5d_6d_7d_8d_9d_{10}$. The last digit,
d_{10}, is a checksum, which is calculated from the other nine digits using the following formula:

$$(d_1 \times 1 + d_2 \times 2 + d_3 \times 3 + d_4 \times 4 + d_5 \times 5 + d_6 \times 6 + d_7 \times 7 + d_8 \times 8 + d_9 \times 9) \ \% \ 11$$

If the checksum is `10`, the last digit is denoted as X according to the ISBN-10 convention. Write a program that prompts the user to enter the first 9 digits and displays the 10-digit ISBN (including leading zeros).

***4.30** (*Hex to decimal*) Write a program that prompts the user to enter a hex character and displays its corresponding decimal integer.

4.31 (*Hex to binary*) Write a program that prompts the user to enter a hex digit and displays its corresponding binary number.

***4.32** (*Decimal to hex*) Write a program that prompts the user to enter an integer between `0` and `15` and displays its corresponding hex number.

***4.33** (*Random strings)* Write a program that generates a random string with three uppercase letters.

***4.34** (*Order three cities*) Write a program that prompts the user to enter three cities and displays them in ascending order.

***4.35** (*Slope-intercept form*) Write a program that prompts the user to enter the coordinates of two points (x1, y1) and (x2, y2) and displays the line equation in the slope-intercept form, i.e.,
y = mx + b. For a review of line equations, see http://www.purplemath.com/modules/strtlneq.htm. m and b can be computed using the following formula:

$$m = (y_2 - y_1)/(x_2 - x_1) \qquad b = y_1 - mx_1$$

Don't display m if it is `1` and don't display b if it is `0` .

***4.36** (*Check SSN*) Write a program that prompts the user to enter a Social Security number in the format ddd-dd-dddd, where *d* is a digit. Your program should check whether the input is valid.

***4.37** (*Generate vehicle plate numbers*) Assume a vehicle plate number consists of three uppercase letters followed by four digits. Write a program to generate a plate number.

Note

More than 100 additional programming exercises with solutions are provided to the instructors on the Instructor Resource Website.