

```

    print(x, y)
def change_us(a, b):
    a = 0
    b = 0
    print(a, b)
main()

```

5. Look at the following function definition:

```

def my_function(a, b, c):
    d = (a + c) / b
    print(d)

```

- Write a statement that calls this function and uses keyword arguments to pass 2 into a, 4 into b, and 6 into c.
 - What value will be displayed when the function call executes?
6. Write a statement that generates a random number in the range of 1 through 100 and assigns it to a variable named `rand`.
7. The following statement calls a function named `half`, which returns a value that is half that of the argument. (Assume the number variable references a float value.) Write code for the function.

```
result = half(number)
```

8. A program contains the following function definition:

```

def cube(num):
    return num * num * num

```

Write a statement that passes the value 4 to this function and assigns its return value to the variable `result`.

- Write a function named `times_ten` that accepts a number as an argument. When the function is called, it should return the value of its argument multiplied times 10.
- Write a function named `get_first_name` that asks the user to enter his or her first name, and returns it.

Programming Exercises

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:

$$\text{Miles} = \text{Kilometers} \times 0.6214$$

2. Sales Tax Program Refactoring

Programming Exercise #6 in Chapter 2 was the Sales Tax program. For that exercise, you were asked to write a program that calculates and displays the county and state sales tax on a purchase. If you have already written that program, redesign it so the subtasks are in functions. If you have not already written that program, write it using functions.



3. How Much Insurance?

Many financial experts advise that property owners should insure their homes or buildings for at least 80 percent of the amount it would cost to replace the structure. Write a program that asks the user to enter the replacement cost of a building, then displays the minimum amount of insurance he or she should buy for the property.

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.

6. Calories from Fat and Carbohydrates

A nutritionist who works for a fitness club helps members by evaluating their diets. As part of her evaluation, she asks members for the number of fat grams and carbohydrate grams that they consumed in a day. Then, she calculates the number of calories that result from the fat, using the following formula:

$$\text{calories from fat} = \text{fat grams} \times 9$$

Next, she calculates the number of calories that result from the carbohydrates, using the following formula:

$$\text{calories from carbs} = \text{carb grams} \times 4$$

The nutritionist asks you to write a program that will make these calculations.

7. Stadium Seating

There are three seating categories at a stadium. Class A seats cost \$20, Class B seats cost \$15, and Class C seats cost \$10. Write a program that asks how many tickets for each class of seats were sold, then displays the amount of income generated from ticket sales.

8. Paint Job Estimator

A painting company has determined that for every 112 square feet of wall space, one gallon of paint and eight hours of labor will be required. The company charges \$35.00 per hour for labor. Write a program that asks the user to enter the square feet of wall space to be painted and the price of the paint per gallon. The program should display the following data:

- The number of gallons of paint required
- The hours of labor required
- The cost of the paint
- The labor charges
- The total cost of the paint job

9. Monthly Sales Tax

A retail company must file a monthly sales tax report listing the total sales for the month, and the amount of state and county sales tax collected. The state sales tax rate is 5 percent and the county sales tax rate is 2.5 percent. Write a program that asks the user to enter the total sales for the month. From this figure, the application should calculate and display the following:

- The amount of county sales tax
- The amount of state sales tax
- The total sales tax (county plus state)

10. 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. Math Quiz

Write a program that gives simple math quizzes. The program should display two random numbers that are to be added, such as:

```
247
+ 129
```

The program should allow the student to enter the answer. If the answer is correct, a message of congratulations should be displayed. If the answer is incorrect, a message showing the correct answer should be displayed.

12. 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.

13. Falling Distance

When an object is falling because of gravity, the following formula can be used to determine the distance the object falls in a specific time period:

$$d = \frac{1}{2} gt^2$$

The variables in the formula are as follows: d is the distance in meters, g is 9.8, and t is the amount of time, in seconds, that the object has been falling.

Write a function named `falling_distance` that accepts an object's falling time (in seconds) as an argument. The function should return the distance, in meters, that the object has fallen during that time interval. Write a program that calls the function in a loop that passes the values 1 through 10 as arguments and displays the return value.

14. Kinetic Energy

In physics, an object that is in motion is said to have kinetic energy. The following formula can be used to determine a moving object's kinetic energy:

$$KE = \frac{1}{2} mv^2$$



VideoNote
The Feet to
Inches Problem

The variables in the formula are as follows: KE is the kinetic energy, m is the object's mass in kilograms, and v is the object's velocity in meters per second.

Write a function named `kinetic_energy` that accepts an object's mass (in kilograms) and velocity (in meters per second) as arguments. The function should return the amount of kinetic energy that the object has. Write a program that asks the user to enter values for mass and velocity, then calls the `kinetic_energy` function to get the object's kinetic energy.

15. Test Average and Grade

Write a program that asks the user to enter five test scores. The program should display a letter grade for each score and the average test score. Write the following functions in the program:

- `calc_average`. This function should accept five test scores as arguments and return the average of the scores.
- `determine_grade`. This function should accept a test score as an argument and return a letter grade for the score based on the following grading scale:

Score	Letter Grade
90–100	A
80–89	B
70–79	C
60–69	D
Below 60	F

16. Odd/Even Counter

In this chapter, you saw an example of how to write an algorithm that determines whether a number is even or odd. Write a program that generates 100 random numbers and keeps a count of how many of those random numbers are even, and how many of them are odd.

17. Prime Numbers

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.



TIP: Recall that the `%` operator divides one number by another and returns the remainder of the division. In an expression such as `num1 % num2`, the `%` operator will return 0 if `num1` is evenly divisible by `num2`.

18. Prime Number List

This exercise assumes that you have already written the `is_prime` function in Programming Exercise 17. Write another program that displays all of the prime numbers from 1 to 100. The program should have a loop that calls the `is_prime` function.

19. Future Value

Suppose you have a certain amount of money in a savings account that earns compound monthly interest, and you want to calculate the amount that you will have after a specific number of months. The formula is as follows:

$$F = P \times (1 + i)^t$$

The terms in the formula are:

- F is the future value of the account after the specified time period.
- P is the present value of the account.
- i is the monthly interest rate.
- t is the number of months.

Write a program that prompts the user to enter the account's present value, monthly interest rate, and the number of months that the money will be left in the account. The program should pass these values to a function that returns the future value of the account, after the specified number of months. The program should display the account's future value.

20. Random Number Guessing Game

Write a program that generates a random number in the range of 1 through 100, and asks the user to guess what the number is. If the user's guess is higher than the random number, the program should display "Too high, try again." If the user's guess is lower than the random number, the program should display "Too low, try again." If the user guesses the number, the application should congratulate the user and generate a new random number so the game can start over.

Optional Enhancement: Enhance the game so it keeps count of the number of guesses that the user makes. When the user correctly guesses the random number, the program should display the number of guesses.

21. Rock, Paper, Scissors Game

Write a program that lets the user play the game of Rock, Paper, Scissors against the computer. The program should work as follows:

1. When the program begins, a random number in the range of 1 through 3 is generated. If the number is 1, then the computer has chosen rock. If the number is 2, then the computer has chosen paper. If the number is 3, then the computer has chosen scissors. (Don't display the computer's choice yet.)
2. The user enters his or her choice of "rock," "paper," or "scissors" at the keyboard.
3. The computer's choice is displayed.
4. A winner is selected according to the following rules:
 - If one player chooses rock and the other player chooses scissors, then rock wins. (Rock smashes scissors.)
 - If one player chooses scissors and the other player chooses paper, then scissors wins. (Scissors cuts paper.)

- If one player chooses paper and the other player chooses rock, then paper wins. (Paper wraps rock.)
- If both players make the same choice, the game must be played again to determine the winner.

22. Turtle Graphics: Triangle Function

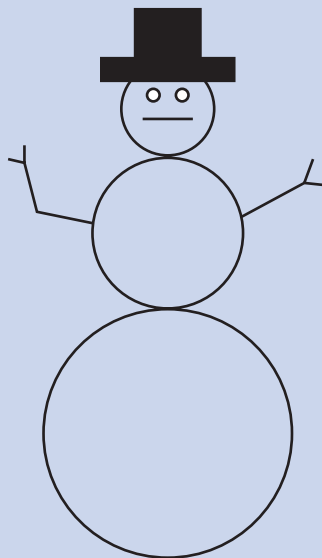
Write a function named `triangle` that uses the turtle graphics library to draw a triangle. The functions should take arguments for the *X* and *Y* coordinates of the triangle's vertices, and the color with which the triangle should be filled. Demonstrate the function in a program.

23. Turtle Graphics: Modular Snowman

Write a program that uses turtle graphics to display a snowman, similar to the one shown in Figure 5-30. In addition to a `main` function, the program should also have the following functions:

- `drawBase`. This function should draw the base of the snowman, which is the large snowball at the bottom.
- `drawMidSection`. This function should draw the middle snowball.
- `drawArms`. This function should draw the snowman's arms.
- `drawHead`. This function should draw the snowman's head, with eyes, mouth, and other facial features you desire.
- `drawHat`. This function should draw the snowman's hat.

Figure 5-30 Snowman



24. Turtle Graphics: Rectangular Pattern

In a program, write a function named `drawPattern` that uses the turtle graphics library to draw the rectangular pattern shown in Figure 5-31. The `drawPattern` function should accept two arguments: one that specifies the pattern's width, and another that specifies the pattern's height. (The example shown in Figure 5-31 shows how the pattern would appear