Exercise Set 2: Simple Functions

Topics:
Patterns: declare-define-use (in the context of functions)
identify components, define components, assemble
components to form a solution: encapsulate
unit conversion pattern
Objects and Functions: using Java2D graphics functions to paint line, rectangle,
ellipse, circle
Java language: function signature/declaration, definition, call, arguments
MathUtilities.randomInt(min, max) function
Exercises: unit conversions, simple drawings

Overview:

These exercises let you practice using and designing simple functions. In the first part
you use existing functions to learn the meaning of the function signature (interface).
Here we selected functions that allow you to do simple graphics - to provide both
motivation and feedback. In the second part you use the declare-define-use pattern
when designing and using new functions. These functions typically encapsulate
computations performed in the previous lesson.

In the previous set of exercises you have done many conversions of measures of
distance, time, temperature, and others. In these exercises we will build utility functions
that will perform the desired conversions.

At the beginning we will do this in two stages. First, we build a function that will
receive the desired data and print the result. We will them modify the function so that it
returns the resulting value to the caller. If we were building a function max(a,b), the
first version would print the maximum, the second version would allow us to use
max(a, b) in computations or to be stored as a value of another variable.

Exercise strategies for students:

In the first part, think carefully about the information you need to supply to the
functions. Observe the different ways in which you can specify the arguments for a
function. Observe also how a function can return a value back to the caller (user).
In the second part, the strategy for designing functions is very similar to the strategy we
used to write our first programs. First specify the computational task the function
should accomplish. Then identify the needed interface: what information needs to be
supplied to the function and what information is the function going to return to the
caller (user). That sets the stage for designing the function signature (the interface).
Finally, you need to define the body of the function that performs the computation and
Exercise Set 2: Simple Functions

design several test cases in which the function is called. These test cases will help you to see that the function works properly.
Exercise Set 2: Simple Functions

Exercises:
Part 1: Solved problems

Defining new functions:
1. Write a function \texttt{drawCircle} that will paint a black circle with the given center and radius.

2. Write a function \texttt{doughnut} that will paint an annulus (doughnut shape). The caller supplies the center of the doughnut and the inside and outside radius of a doughnut. Paint the outside circle and invert the inside circle.

3. Write the function \texttt{doughnut2} that will paint the doughnut. The caller supplies the center of the doughnut, the outside radius and the thickness of a doughnut. Paint the outside circle and invert the inside circle.

4. Define a function \texttt{PaintWindow}, given its top left corner, its height and its width. It is a framed rectangle with a vertical and horizontal line through the middle. You may choose to paint the rectangle with some color, then frame it in black.

5. Write a function \texttt{CountCoins1} that will use four arguments representing the number of quarters, dimes, nickels, and pennies and will print total amount in cents.

6. Write a function \texttt{CountCoins2} (a modification of \texttt{CountCoins1}), so that it will return the amount as a function value. Write a program that will use this function to print the amount in dollars and cents.

7. Write the function \texttt{ConvertToMinutes} that will help you figure out how long a football game lasted. The user will supply the hour and minute when the game started and the hour and minute when it ended. We assume that it ends before midnight of the day on which it began. To make things easier, the time will be given in military time (i.e. 24 hour clock).

   The function will take the time in hours and minutes and convert it to the total minutes elapsed since the beginning of the day.

   Write a program that will test this function.
Exercise Set 2: Simple Functions

Part 2: Practice Problems - Easier

Designing new functions:

8. Define a function `ScaryEye`. It requires four arguments, the coordinates of two corners of the eye. The eye is a black rectangle with a red oval inset 2 pixels. Write a program that will test this function.

9. Define a function `Diamond`. It requires three arguments: the coordinates of its center and its width. Write a program that will test this function.

10. Write a function `NumberOfPages` that will count how many pages long is a given document. The function arguments will be the word count, the average number of words per line and the number of lines per page. Write a program that will test this function. Find out real values for the input data - for single spaced and double spaced pages, for small announcements, etc.

11. Write a function `CountBills1` that will use four arguments representing the number of $20 bills, $10 bills, $5 bills and $1 bills and will print the total amount in dollars.

12. Write a function `CountBills2` (a modification of `CountBills1`), so that it will return the amount as a function value. Write a program that will read from the user the entire list of currency (both bills and coins) in his wallet and print the total amount in dollars and cents.

13. Write a function `MinutesPerMile` that will aid a runner in monitoring the training. The function arguments specify the length of the course in miles and the time the runner took to run the course, given in minutes and seconds. Return the value in minutes and decimal fractions of a minute (one double value). Write a program that will test this function.

14. Write a function `GrossPay` that will compute the gross monthly salary for a salaried employee. The function arguments specify the yearly base salary and how many percent of salary is deducted for retirement contributions (this amount is not included in the gross pay). Write a program that will test this function.
Part 3: Practice Problems - Harder

15. Recently a new World record for 400m has been set at 43.18 seconds. Write a function that will read the length of the race in meters and the time in seconds and fractions of seconds and return the average speed of the racer in miles per hour.

To make your life easier, start by writing a function that will convert the distance in meters into the distance in miles. Write another function that will convert the time in seconds and fraction of seconds into hours. Then write the final function. Write a program that will test your whole design.

16. One pound of flour equals four cups. Write a function for a cook, so she can enter as a function argument the number of cups needed and the function will return as arguments the amount in pounds and ounces. Allow the user to enter 1.5 cups or 0.666 cups. Document your strategy and write a program that will test this function.

17. Write a function MakeChange. The user needs to supply one argument that specify the amount in the range from 1 to 99. (The function does not need to verify that the input is correct.) The function print the number of quarters, dimes, nickels, and pennies needed to make the desired amount.

18. Grocery stores are required to post unit price for cereal they sell. Write the program that will compute the unit price using two helping functions. The first function will convert the weight of the box from pounds and ounces to ounces. The second function will compute the unit price, using the weight (in pounds and ounces) and price in dollars (recorded with decimals) as function arguments.

19. Write a program that will paint a tree. The use supplies the center of the base of the trunk of the tree, the trunk height and the radius of its crown. The width of the tree trunk will be the larger of 1/4 of the trunk height or 5. Paint the trunk black and the crown green. Add 2 pixels to the trunk height, so that it reaches the entire bottom of the crown.

20. Define a function PaintDoor. Here you may supply the bottom left point instead of the center. Door is a rectangle, with a circle window (maybe in white) centered at 2/3 of the height with radius 1/4 of the width.

21. For a home remodeling project you want to tile a floor with new tiles. Write a function that will use as arguments the dimensions of the room (assume it is a rectangle) and the size of the square tiles to be used. The function will return a count how many tiles will be needed. Of course, the tiles at two sides may need to be cut to fit. Write a program to test this function.

22. Modify the previous function to compute the number of boxes of 12 tiles that will be needed. You cannot buy less than a full box of 12 tiles. Write a program that uses this function and also asks the user to type in the cost of a box of tiles and will print the total cost of needed tiles.
Exercise Set 2: Simple Functions

23. Write a function `TimeToDownload` that will use two arguments specifying the size of a picture to download, and the transfer rate given as number of bytes per second. The function will return the time needed to transfer the picture, in minutes rounded to the nearest highest minute.

24. Write two functions that will aid an overseas traveler: `ConvertFromDollars` and `ConvertToDollars`. The first two arguments for each of these functions will be the conversion rate x dollars = y amount of foreign currency. The third argument will be the amount of dollars or foreign currency we want to convert. Write a program to test your functions.

What would you expect the following statement to print?

```java
console.out.println(ConvertToDollars (x, y, ConvertFromDollars (x,y, 100));
```