

Writing Methods

For each exercise below, write the method described. Assume all ranges are inclusive (include both end points).

1. Write a method called **powersOfTwo** that prints the first 10 powers of 2 (starting with 2). The method takes no parameters and doesn't return anything.
2. Write a method called **alarm** that prints the word "Alarm!" multiple times on separate lines. The method should accept an integer parameter that specifies how many times the output line is printed.
3. Write a method called **sum100** that returns the sum of the integers from 1 to 100.
4. Write a method called **sumRange** that accepts two integer parameters that represent a range. You may assume the first parameter is less than or equal to the second. The method should return the sum of the integers in that range.
5. Write a method called **maxOfTwo** that accepts two integer parameters and returns the larger of the two.
6. Write a method called **larger** that accepts two floating point parameters (of type `double`) and returns true if the first parameter is greater than the second, and false otherwise.
7. Write a method called **evenlyDivisible** that accepts two integer parameters and returns true if the first parameter is evenly divisible by the second, or vice versa, and false otherwise. You may assume that neither parameter is zero.
8. Write a method called **average** that accepts three integer parameters and returns their average as a floating point value.
9. Write a method called **isAlpha** that accepts a character parameter and returns true if that character is either an uppercase or lowercase alphabetic letter.
10. Write a method called **validate** that accepts three integer parameters. The first two parameters represent a range, and the purpose of the method is to verify that the value of the third parameter is in that range. You may assume that the first parameter is less than or equal to the second. If the third parameter is not in the specified range, the method should prompt the user and read a new value. This new value should be tested for validity as well. The method should only return to the calling method once a valid value has been obtained, and it should return the valid value.
11. Write a method called **floatEquals** that accepts three floating point values as parameters. The method should return true if the first two parameters are essentially equal, within the tolerance of the third parameter.
12. Write a method called **isIsocoles** that accepts three integer parameters that represent the lengths of the sides of a triangle. The method should return true if the triangle is isosceles but not equilateral, meaning that exactly two of the sides have an equal length, and false otherwise.
13. Write a method called **randomInRange** that accepts two integer parameters representing a range. You may assume that the first parameter is less than or equal to the second, and that both are positive. The method should return a random integer in the specified range.