# Module 6.2 Function Basics: Coding Questions with Hints

## Question 1: Define a function `square` that returns the square of its argument.

Hint: Use a single parameter for the function and return the parameter multiplied by itself.

## Question 2: Write a function `greet\_user` that takes a name as an argument and prints a greeting.

Hint: The function should include a parameter for the name and use it in the greeting printed.

## Question 3: Create a function `calculate\_area` that computes the area of a rectangle.

Hint: The function should take two parameters (length and width) and return the product of these.

## Question 4: Implement a function `is\_odd` that returns whether a number is odd.

Hint: Check if the number is odd by using the modulus operator `% 2` and return the boolean result.

## Question 5: Define a function `list\_sum` that returns the sum of elements in a list.

Hint: Iterate over each element in the list to calculate the sum and return it.

## Question 6: Write a function `contains\_element` that checks if an element exists in a list and returns a boolean.

Hint: Use the `in` keyword to check for the element's presence in the list.

## Question 7: Create a function `add\_to\_list` that adds an element to a list and returns the list.

Hint: The function should take two parameters: the list and the element to add. Use the list's `append` method.

## Question 8: Implement a function `get\_average` that calculates the average of numbers in a list.

Hint: Sum up the elements using a loop or the `sum` function and divide by the length of the list.

## Question 9: Define a function `reverse\_list` that takes a list and returns it reversed.

Hint: Use list slicing or a loop to create and return the reversed list.

## Question 10: Write a function `multiply\_numbers` that takes an arbitrary number of arguments and returns their product.

Hint: Use the `\*` operator to accumulate the product of the arguments. Consider using a loop to iterate over the arguments.