

```

1 import java.util.Scanner;
2
3 public class Functions {
4
5     //Parts of a function heading
6     /*
7     * public - access modifier Basically it means this can be run from any part of the package
8     * static - Means there is only ever one instance of this class object (OOP - more on this later)
9     *         - needs to be here in classes with main methods
10    *         - separates functions from something called (Methods OOP)
11    * return type (what data will this function pass out)
12    *         - void for no data
13    *         - String or any other object
14    *         - Any primitive
15    * variable - the name of this function
16    * (parameters) - the data a function needs to run
17    */
18    public static void sayHi() {
19        System.out.println("Hello");
20    }
21
22    public static void sayHiTo(String name) { //function with parameter (data passed in)
23        System.out.println("Hello " + name + ".");
24    }
25
26    public static String askQuestion(String question) { //using String instead of void means the function will pass a String out
27        Scanner scan = new Scanner(System.in);
28        System.out.println(question);
29        return scan.nextLine(); // return passes data out of a function and terminates the function
30        //System.out.println("Does not matter"); //unreachable
31    }
32
33    public static boolean stringContains(char letter, String word) { // Two params checking if a letter is in a word
34        for(int i = 0; i < word.length(); i++) {
35            if(word.charAt(i) == letter) { // if it finds the letter it returns true
36                return true;
37            }
38        }
39        return false; //has not found the letter yet so it returns false
40    }
41
42
43
44    public static void main(String[] args) {
45        sayHi();
46        sayHi();
47        sayHiTo("Bob");
48        sayHiTo("Sarah");
49        String name = askQuestion("What is your name?"); //saves the result of the function into name
50        sayHiTo(name); // uses name in another variable
51        System.out.println(stringContains('o', "This word"));
52        System.out.println("This example".contains("This"));
53        //String.contains(String) already exists and returns a boolean if it finds the substring
54
55    }
56
57 }
58

```