

1a. not a compile time error

1b. not a compile time error

1c. static semantic (contextual) error

1d. static semantic (contextual) error

1e. static semantic (contextual) error

1f. not a compile time error

1g. not a compile time error

1h. not a compile time error

1i. static semantic (contextual) error

1j. not a compile time error

1k. syntax error

1l. syntax error

1m. not a compile time error

2. JavaScript would throw a syntax error because variables declared with 'let' cannot be redeclared within the same scope. In Rust, the second declaration 'shadows' the first one meaning it takes precedence over the first one.

3. In Java, private methods can be called explicitly within the defining class whereas private methods in Ruby must be called implicitly. Private methods in Ruby apply at the instance level, not the class level.

4. It is possible that different evaluation orders can lead to different arguments being passed. If we had a function `function1()` that accepted a global variable as an argument and another function as an argument, the call `function1(function2(), a)` could be different from `function1(a, function2())` if `function2` were to modify `a`.

5. The Carlos language does not allow recursive structs because it would lead to an infinitely-sized structure. No field of the struct is allowed to have the type of the struct itself. If the compiler finds that any fields in the struct share the same type as the struct itself, an error will be thrown.

6. Python:

```
import numpy as np
def minimum_value(num_list, min_num = np.inf):
    if not num_list:
        return min_num
    return minimum_value(num_list[1:], min(min_num, num_list[0]))
```

C:

```
double minimum_value(double *array, int len, double min_num) {
    if (len == 0) {
        return min_num;
    }
    double new_num = array[0] > min_num ? min_num : array[0];
    return minimum_value(array + 1, len - 1, new_num);
}
```

JavaScript:

```
function minimumValue(array, minNum = Infinity) {
    if (array.length == 0) {
        return minNum;
    }
```

```

    }
    return minimumValue(array.slice(1), Math.min(minNum, array[0]));
}

```

Go:

```

func minimumValue(array []float64, minNum float64) float64 {
    if len(array) == 0 {
        return minNum
    } else {
        newNum := minNum
        if minNum > array[0] {
            newNum = array[0]
        }
        return minimumValue(array[1:], newNum)
    }
}

```

7. The second friend's function is passing, `update(i)`, which is an undefined value, to `setTimeout`.

`Update(i)` is undefined because there is no return value specified in the function. `Update(i)` is being called immediately. To fix this, they should change it to `() => update(i)` because it ensures that it is called after 1000 milliseconds go by instead of before.

8. Linter errors (SonarQube):

`import`: Move this file to a named package.

`m`: Variable `m` is never read.

`HashMap<String, Integer>`: Redundant type arguments in new expression (use diamond operator instead).

`zero`: `zero` is never used.

0: Remove this method and declare a constant for this value.

C: Add a nested comment explaining why this method is empty, throw an `UnsupportedOperationException` or complete the implementation.