Posed Questions for Duck Typing Across Languages: Python, JavaScript, Java, and C#

[Link](Duck typing formal definition) to the ChatGPT chat

Introduction

- I am a student of Master degree in computer science, and I am attending the course of advanced programming, which focuses on subtle aspect of programming, such as memory management, scoping rules, typing rules, polymorphism and so on. My final project must analyse how dynamically typed languages (Python, JavaScript) rely on duck typing, and compare this with the more static, nominal type systems of Java and C#, examining where and how duck typing–like mechanisms (interfaces, reflection, dynamic features) can appear in statically typed languages. I would like to start by understanding formally how duck typing can be defined.
- Before going on, I would like to know why it was introduced, the problems it meant to solve, and so its main benefits and drawbacks.
- Can you also show me the distinction between nominal typing, statical structural typing and duck typing?

Dynamically typed languages

- Why dynamically typed languages such as Python and JavaScript rely on duck typing?

Python

- Python bases its type system on duck typing. Explain and show to me how the runtime of Python actually implements duck typing, providing examples and using advanced programming concept.
- I want to know better what a type object is and how the dynamic lookup for method accesses is actually performed.
- For what concerns the step-by-step lookup process, I want to know how the check in the instance dictionary and in the class object are performed, particularly if some syntactic sugar is used or if a method invocation is directly interpreted using PyObject and PyTypeObject references.
- Show me the C implementation for PyObject_GenericGetAttr, explaining the role of important variables such as "f" and "descr", and explaining precisely the order of the performed lookup (inside the object, inside the class, inside the inherited hierarchy...)
- Now the steps are more clear, but i want to know why the dictionary lookup follows the class lookup, when at high level the instance implementation of methods is prioritized over the class implementation.
- now can you provide some code example of interesting use cases of duck typing in python?

JavaScript

- Now i want to focus on the JavaScript language. Show me how duck typing is achieve in JavaScript, explaining precisely the steps performed at runtime, with code examples.
- I want a deeper view in the object model of JavaScript and on how the lookup is actually performed.
- When I use the new keyword to create a new class instance, what is the shape of the object and what operations are performed
- Try showing me some low-level implementation for the Get and Construct methods.
- Yes, but also explain what are proxy objects and make some code examples.
- Now to conclude this part, can you provide some code example of interesting use cases of duck typing in javascript? possibly reuse and adapt the example you generated for the python analysis.

Statically typed languages

- Now I want to shift the focus to statically typed languages. Why do statically typed languages rely on explicit type systems and compile-time checks?

Java

- Now I want to focus on statically typed languages, in particular Java. I want to know how type compatibility is ensured, particularly when performing attribute access or method invocations
- Can you show me a lower level perspective of the process behind a method invocation?
- Can you explain better how the method invocation is performed, focusing on compile-time operations such as class linking and constant pool generation, but also on runtime aspect, such as the resolution of symbolic method references?
- How can Java support more duck typed behaviours with interfaces and reflection API?

C#

- Now I want to analyse the C# programming language. I want to know its type system works, particularly when performing method invocations and attribute access.
- Can you show me from a lower view how method invocation is performed in C#?
- C# support both interfaces and reflections. Can you provide me some code examples that tries mimicking ducked typed behaviours in C#?
- Can you explain me the usage of the dynamic keyword in C#?