The program in Fig-1 is implemented in (simplified) Java. The program in Fig-2 is implemented in Python. Do these languages implement inheritance in the same way?

```java
public class Zoo {
  class Animal {
    public void eat() {
      System.out.println("Eats.");
    }
  }

  class Mammal extends Animal {
    public void suckMilk() {
      System.out.println("Baby.");
    }
  }

  public void test (boolean t) {
    Animal a = new Animal();
    if (t) {
        a = new Mammal();
    }
    a.eat();
  }
}
```

Figure 1: Dynamic dispatch in Java

```python
class Animal:
  def eat(self):
    print("Eats.")

class Mammal(Animal):
  def suckMilk(self):
    print("Baby.")

def test(t):
  a = Animal()
  if t:
    a = Mammal()
  a.eat()
```

Figure 2: Dynamic dispatch in Python

UFMG
DCC
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO
Compilers Laboratory

1 What is dynamic dispatch?

2 How could the calls to the method `eat` be implemented in these two languages? (think as if you were compiling them to some simple assembly code)

3 Does a call to `eat` take the same assymptotic time to be resolved in these two programs? Or do these times differ? (The question is not about the time to process `eat()`; rather, it's about the computational cost to find the correct implementation of it, e.g., is it O(1), O(...)?)