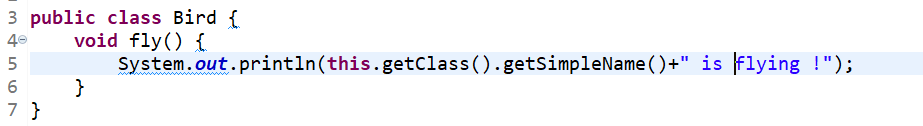
The Liskov Substitution Principle states that subclass should be substitutable for their parent class.

First look at the below code to get an idea about LSP,

We have a parent class called **Bird**



So who can inherit this Bird class

* Sparrow.java
* Crow.java
* Owl.java
* Parrot.java

All the above classes can surely extends the Bird class

**But, can Ostrich.java class extends the Bird class ?**

Answer is **No.**

**Explanation :** This is because Bird.java has method called fly() in it. But if Ostrich.java extends the Bird class then that fly() method will also be available in Ostrich.java that will break the LSP. Because Ostrich can’t fly.

So, in this condition Ostrich has to avoid the fly() method by which Ostrich.java won’t be substitutable in place of Bird.java

In below picture, the Sparrow class is extending Bird class that is fine because sparrow can fly.

Graphical user interface, text, application

Description automatically generated

But in this below picture, Ostrich is extending the Bird that is not fine because ostrich cannot fly.

Text, letter

Description automatically generated

Our main class

