

# Roadmap For Mastering OOP

# OOP Principles

The **SOLID Principles** are a good place to start.

Some are more important than others at the beginning.

The Single Responsibility Principle is most important.

Others, like the Liskov Substitution principle, are critical to unlocking higher levels of mastery.

# More Intricate Inheritance

We showed you the useful basics of inheritance already.

There is much more to learn and exploit with this principle.

Ranging from the cognitive skills of how you mentally represent complex inheritance hierarchies...

To how you can "jump up" the inheritance chain in subclasses, and the idioms and patterns for that...

To how to write generic code that works on a range of related classes.

# Design Patterns

When writing object-oriented code, it is important to leverage the tried and tested design patterns.

The standard design patterns apply to Python...

But they apply DIFFERENTLY. Because as a language, Python's feature set is different.

# Properties

A kind of a hybrid between a member variable and a method.

Used for validation, refactoring, and instance state management.

Other OO languages have this...

But in Python, it works differently than you have ever seen before.

# Access Control

This refers to who can access attributes of the instance...

Who can access is member variables, or invoke its methods.

Every mainstream OO has this, because it is useful.

But again, it works differently in Python than in any other language.

# Abstract Classes and Interfaces

As your application grows in size and complexity...

How do you enforce "contracts" new classes must fulfill?

This is what abstract classes and interfaces are for.

Yet again, the tools for this in Python work completely differently than in any other language.

# Other Python-Specific Differences

How to work around the lack of multiple dispatch in Python...

No implicit or explicit interfaces...

The "Single constructor" problem and its solutions...

And more.

Every language has its idiosyncrasies. And it is important to grasp what they are... how to leverage their benefits, and work around their downsides.