**Python -Beyond The Basic**

# Prerequisites

## Prerequisites

# Organizing Larger Programs

## Packages

## Imports from sys.path

## Implementing packages

## Subpackages

## Example: A full program

## Relative imports

## Controllings imports with\_\_all\_\_

## Namespaces packages

## Executable directories

## Recommended layout

## Duck tails: modules are singletons

## summary

# Beyond Basic Functions

## Function Review

## Callable instances

## Classes are callable

## Conditional expressions

## Lambdas

## Detecting callable objects

## Extended Formal Argument Syntax

## Extended call syntax

## Forwarding arguments

## Duck tail: transposing tables

## summary

# Closures and Decorators

## Local functions

## Returning functions from functions

## Closures and nested scopes

## Function factories

## The nonlocal keyword

## Function decorators

## A first decorator example

## What can be a decorator?

## Instances as decorators

## Multiple decorators

## Decorating methods

## Functools.wraps()

## Duck tails: validating arguments

## Summary

# Properties and Class Methods

## Class attributes

## Static methods

## Class methods

## Static methods with inheritance

## Class methods with inheritance

## Properties

## Properties and inheritance

## Duck tail: the template method pattern summary

# String and Representations

## Two string representations

## Repr()

## Str()

## When are the representations used ?

## Interaction with format()

## Reprlib

## Asciii(), ord(), and chr()

## Duck tail: bigger isn’t always better

## Summary

# Numeric and Scalar Types

## Reviewing int and float

## The decimal module and the decimal type

## Rational numbers with the fraction type

## The complex type and the cmath module

## Built-in numeric function abs() and round()

## Number base conversions

## The datetime module and date type

## The time type

## The datetime type

## Durations with the timedelta type

## Arithmetic with datetiem

## Time zones

## Duck tail: floating point vs rational numbers

## Summary

# Iterables and Iteration

## Multi-Input Comprehensions

## Nested Comprehensions

## The map() Function

## Multiple Input Sequences

## map() Versus Comprehensions

## The filter() Function

## The functools.reduce() Function

## Combining map() and reduce()

## The Iteration Protocols

## Putting the Protocols Together

## Alternative Iterable Protocol

## Extended iter() Format

## Duck Tail: Iterator for Streamed Sensor Data

## Summary

# Inheritance and subtype Polymorphism

## Inheritance Overview

## A Realistic Example: SortedList

## The Built-In isinstance() Function

## The Built-In issubclass() Function

## Multiple Inheritance

## Details of Multiple Inheritance

## Method Resolution Order

## How Is Method Resolution Order Calculated?

## The Built-In super() Function

## Class-Bound Super Proxies

## Instance-Bound Super Proxies

## Calling super() Without Arguments

## SortedIntList Explained

## The Object Class

## Duck Tail: Inheritance for Implementation Sharing

## Summary

# Implementing Collections

## Collection Protocol Overview

## Collection Construction

## The Container Protocol

## The Sized Protocol

## The Iterable Protocol

## The Sequence Protocol: Indexing

## The Sequence Protocol: Slicing

## Comprehensible Test Results With \_\_repr\_\_()

## Implementing Equality and Inequality

## The Sequence Protocol: Reversing

## The Sequence Protocol: index()

## The Sequence Protocol: count()

## Improving Performance from O(N) to O(log n)

## Refactoring to Avoid Don't Repeat Yourself (DRY)

## Checking Protocol Implementations

## The Sequence Protocol: Concatenation and Repetition

## The Set Protocol

## Duck Tail: Making a Mutable Set

## Summary

# Exceptions and Errors

## Always Specify an Exception Type

## The Standard Exception Hierarchy

## Exception Payloads

## Defining New Exceptions

## Chaining Exceptions

## Traceback Objects

## Assertions: Internal Invariants

## Assertions: Class Invariants

## Assertions: Performance

## Duck Tail: Preconditions and Postconditions

## Summary

# Defining Context Managers

## What Is a Context Manager?

## The Context Manager Protocol

## A First Context Manager Example

## \_\_enter\_\_()

## \_\_exit\_\_()

## \_\_exit\_\_() and Exception Propagation

## The with-statement Expansion

## contextlib.contextmanager

## Multiple Context Managers

## Don't Pass a List!

## Duck Tail: Context Managers for Transactions

## Summary

# Introspection

## Object Types in Depth

## Introspecting Objects

## Introspecting Scopes

## The Python Standard Library Inspect Module

## Duck Tail: An Object Introspection Tool

## Summary