Sharif University of Technology
Department of Computer Engineering

Fundamentals of Programming

Python Language

Arman Malekzadeh
PhD Candidate in Artificial Intelligence

Table of contents

1 Class Inheritance

Class Inheritance

Class Inheritance

- Inheritance is a way of creating a new class for using details of an existing class without modifying it.
- The newly formed class is a derived class (or child class). Similarly, the existing class is a base class (or parent class).
- The idea of inheritance implements the IS-A relationship.
- It also provides reusability of a code. We don't have to write the same code again and again. Also, it allows us to add more features to a class without modifying it.

Class Inheritance: Terminology

- Base Class (Parent Class): The class whose properties are inherited is known as a base class (or a parent class or a superclass).
- Derived Class (Subclass): The class that inherits the properties from another class is known as a derived class (or a child class or a subclass).
- Syntax:

```
class BaseClass:
    # Base class code

class DerivedClass(BaseClass):
    # Derived class code
```

Class Inheritance: Super Function

- The super() function is used to give access to methods and properties of a parent or sibling class.
- The super() function returns an object that represents the parent class.
- Syntax:

```
class DerivedClass(BaseClass):
    def __init__(self, additional_param):
        super().__init__()  # Call the constructor of the base class
        # Derived class initialization code
```

Class Inheritance: Method Overriding

- Method overriding is an ability of any object-oriented programming language that allows a subclass or child class to provide a specific implementation of a method that is already provided by one of its super-classes or parent classes.

- Syntax:

```
class BaseClass:
    def some_method(self):
        print("Base class method")

class DerivedClass(BaseClass):
    def some_method(self):
        print("Derived class method")
```

Class Inheritance: Multiple Inheritance

- Multiple inheritance is a feature of some object-oriented computer programming languages in which an object or class can inherit characteristics and features from more than one parent object or parent class.

- Syntax:

```
class BaseClass1:
    # Base class 1 code

class BaseClass2:
    # Base class 2 code

class DerivedClass(BaseClass1, BaseClass2):
    # Derived class code
```

Class Inheritance: Example 1

```python
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def print_info(self):
        print(f"Name: {self.name}, Age: {self.age}")

class Student(Person):
    def __init__(self, name, age, student_id):
        super().__init__(name, age)
        self.student_id = student_id

    def print_info(self):
        super().print_info()
        print(f"Student ID: {self.student_id}")
```

Class Inheritance: Example 2

```
class Rectangle:
    def __init__(self, width, height):
        self.width = width
        self.height = height

    def area(self):
        return self.width * self.height

class Square(Rectangle):
    def __init__(self, side):
        super().__init__(side, side)
```

Class Inheritance: Example 3 (Place and Home)

```python
class Place:
    def __init__(self, name, address):
        self.name = name
        self.address = address

    def print_info(self):
        print(f"Name: {self.name}, Address: {self.address}")

class Home(Place):
    def __init__(self, name, address, area, rooms):
        super().__init__(name, address)
        self.area = area
        self.rooms = rooms

    def print_info(self):
        super().print_info()
        print(f"Area: {self.area}, Rooms: {self.rooms}")
```

References

References I

[1] B Downey, A. (2015). Think Python: How to Think Like a Computer Scientist-2nd Edition.

[2] Deitel, H. M., & Deitel, P. J. (2004). C: How to program. Pearson Educacion.

Sharif University of Technology
Department of Computer Engineering

Arman Malekzadeh

Fundamentals of Programming

Python Language