تكليف

Multiple Inheritance and Built-in Modules in Python

اعداد

نادر فايز قاسم علي سعيد

Task 1: Multiple Inheritance in Python

Multiple inheritance is a concept in object-oriented programming that allows a class to inherit features and methods from more than one parent class. This helps in code reuse and combining functionalities from different sources into a single class.

Advantages:

- Code reuse and reduced duplication.
- Combining behaviors from multiple classes.
- Flexibility in software design.

Example 1: Basic Multiple Inheritance

```
class Father:
    def skill_father(self):
        print("Father knows driving.")

class Mother:
    def skill_mother(self):
        print("Mother knows cooking.")

class Child(Father, Mother):
    pass

c = Child()
c.skill_father()
c.skill_mother()
```

Example 2: Conflict Resolution using MRO

```
class A:
    def show(self):
        print("Class A")

class B:
    def show(self):
        print("Class B")

class C(A, B):
    pass
```

```
obj = C()
obj.show()

print(C.__mro__)
```

Python uses Method Resolution Order (MRO) to determine the order in which parent classes are searched when a method is called.

Task 2: Built-in Packages and Modules

Built-in modules in Python are libraries that are already available and do not require installation. They provide ready-to-use functions to simplify programming.

- math: Used for mathematical calculations.
- random: Used for generating random numbers.
- datetime: Used for handling date and time.
- os: Used to interact with the operating system.
- sys: Used to interact with the Python runtime environment.

Examples:

```
import math
print(math.sqrt(25))

import random
print(random.randint(1, 10))

import datetime
print(datetime.datetime.now())
```

These modules improve productivity, reliability, and performance because they are optimized and tested.