Course	
Term	
Week	
Date	
Chapter. Topic	10. Classes and Object-Oriented Programming

Classes and DUNDER methods

Siva R Jasthi

Computer Science and Cybersecurity

Metropolitan State University

Student class

- student1 = Student('Mark', 'mark@gmail.com', 100)
- student2 = Student('Alex', 'alex@yahoo.com', 100)

```
X = 10
Y  = 20
print(x == y)
```

```
print(student1 == student2)
```

==, !=, <, >, <=, >= → these definitions (meanings are dictated by you)

Dunder (**D**ouble **Under**score) Methods

- In Python, "dunder" methods are special methods that have double underscores on both sides of their names, such as __init__, __repr__ and __str__.
- These methods are also called "magic methods" or "special methods"
- These are used to define how objects of a class behave in various situations.

Dunder (**D**ouble **Under**score) Methods

- __init__(self, ...): The constructor method, used to initialize object attributes when an instance of the class is created.

- __str__(self): This method defines the "informal" or user-friendly string representation of an object. It is called by the str() function and is often used for debugging and display purposes.

- __repr__(self): This method defines the "formal" or unambiguous string representation of an object. It is called by the repr() function and is typically used for debugging and development.

Dunder Methods for comparison

- __eq__(self, other): Defines the behavior of the equality operator (==) when comparing two objects of the same class.
- __ne__(self, other): Defines the behavior of the inequality operator (!=) when comparing two objects of the same class.
- __lt__(self, other): Defines the behavior for the less than operator (<) when comparing two objects of the same class.
- __le__(self, other): Defines the behavior for the less than or equal to operator (<=) when comparing two objects of the same class.
- __gt__(self, other): Defines the behavior for the greater than operator (>) when comparing two objects of the same class.
- __ge__(self, other): Defines the behavior for the greater than or equal to operator (>=) when comparing two objects of the same class.

Dunder (**D**ouble **Under**score) Methods

- __len__(self): Used to define the behavior of the len() function for objects of the class. This method is typically implemented for sequences and collections.
- __getitem__(self, key): Defines how an object is accessed using square brackets (e.g., obj[key]). This method is used for indexing.
- __setitem__(self, key, value): Defines how an object is modified when using square brackets (e.g., obj[key] = value).
- __delitem__(self, key): Defines the behavior when an item is deleted using the del statement with square brackets (e.g., del obj[key]).

Dunder (**D**ouble **Under**score) Methods

- __iter__(self): Used to define how an object can be iterated, typically by returning an iterator object.

- __next__(self): Defines the behavior of retrieving the next item in an iterator when using the next() function.

Dunder (**D**ouble **Under**score) Methods

- __contains__(self, item): Defines the behavior when using the in operator to check if an item is present in an object.

- __del__(self): The destructor method, used to perform cleanup operations before an object is destroyed.

Summary

“Object Oriented Thinking” – the concepts we talked today are not specific to Python.

These are valid and applicable in any programming language.

The syntax might differ between the languages. However, the foundational concepts are common.

Get started with solid understanding on “Objects and Classes”!

We just covered C of Objects in CRUD. We will review R,U, and D in the next class.