



Python Basics: Classes, Exception Handling

TAKEAWAYS

Classes and Objects

- 1** Object-oriented programming (OOPS) makes managing software projects easier.
- 2** Classes and objects are at the core of OOPS.
- 3** Classes are like a blueprint of an entity whereas objects represent a specific instance of that class. For example, Human is a class whereas specific human being (say Mira Sharma) is an instance of the class Human.
- 4** Classes have properties and functions.
- 5** Use the `__init__` method to initialize object properties when an object is created.
- 6** Methods defined in a class describe the behaviors that objects of that class can perform.

Operator Overloading

- 1** Operator overloading allows you to define how operators like `<`, `>`, `==`, `+`, `-`, `+`, `-`, `*` etc. behave for custom objects.
- 2** Overloading operators makes your custom classes more intuitive and easier to use.
- 3** Operator overloading enhances the readability and maintainability of your code by allowing objects to interact naturally with built-in operators.

Inheritance

- 1** Inheritance encourages a structured approach to code organization, promoting clarity and maintainability.
- 2** Inheritance allows new classes to absorb attributes and methods from existing classes, enhancing code reusability.
- 3** It facilitates the extension and customization of existing code by deriving new subclasses.

Exception Handling

- 1 Exception handling ensures that programs can address and recover from errors during execution without crashing.
- 2 Using **try** and **except** blocks allows developers to separate normal code from error handling code, enhancing readability.
- 3 Specifying particular exceptions to catch enables targeted responses to different error types, improving error resolution effectiveness.
- 4 The **finally** block executes code regardless of whether an exception was raised or not, ensuring that cleanup and release of resources can always occur.

`__main__` Function

- 1 if `__name__ == "__main__"` is a way to define entry point in Python code (similar to main function in C++ or JAVA)
- 2 When you import any module (e.g. `import numpy as np`), the `module.__name__` is set to the name of that module (e.g. `np.__name__` is set as 'numpy')