## 1. Circle Class

python

Copy

import math

class Circle:

def \_\_init\_\_(self, radius):

self.radius = radius

def area(self):

return math.pi \* self.radius \*\* 2

def perimeter(self):

return 2 \* math.pi \* self.radius

# Example usage

circle = Circle(5)print("Area:", circle.area())print("Perimeter:", circle.perimeter())

## 2. Class without ****init****()

True. We can define a class without the \_\_init\_\_() function, but it won't be able to initialize instance attributes automatically when an object is created.

## 3. Rectangle Class

python

Copy

class Rectangle:

def \_\_init\_\_(self, length, width):

self.length = length

self.width = width

def area(self):

return self.length \* self.width

def perimeter(self):

return 2 \* (self.length + self.width)

# Example usage

rect = Rectangle(4, 6)print("Area:", rect.area())print("Perimeter:", rect.perimeter())

## 4. Square Class

python

Copy

class Square:

def \_\_init\_\_(self, side):

self.side = side

def area(self):

return self.side \*\* 2

def perimeter(self):

return 4 \* self.side

# Example usage

square = Square(5)print("Area:", square.area())print("Perimeter:", square.perimeter())

## 5. Use of ****init****()

The \_\_init\_\_() function is a special method in Python classes that is automatically called when an object is created. It's used to initialize the object's attributes.

## 6. Purpose of ****str****()

The \_\_str\_\_() function is used to return a human-readable string representation of an object. It's called by the str() built-in function and by the print() function.

## 7. Attributes of an Object

Attributes are variables that belong to an object and represent its state or characteristics. For example, in a Car class, attributes might include color, model, and speed.

Example:

python

Copy

class Car:

def \_\_init\_\_(self, color, model):

self.color = color

self.model = model

my\_car = Car("red", "Toyota")print(my\_car.color) # "red" is an attributeprint(my\_car.model) # "Toyota" is an attribute

## 8. Parent Class and Subclass Names

Parent class is also called superclass or base class. Subclass is also called derived class or child class.

## 9. Purpose of super()

The super() function is used to call methods from a parent class in a subclass, allowing for method overriding while still maintaining access to the parent class's functionality.

## 10. Types of Inheritance

Single Inheritance: A subclass inherits from only one superclass.

Multiple Inheritance: A subclass inherits from multiple superclasses.

Multilevel Inheritance: A subclass inherits from another subclass.

Hierarchical Inheritance: Multiple subclasses inherit from a single superclass.

Hybrid Inheritance: A combination of multiple inheritance types.

## 11. Private vs Protected Members

Private members (prefixed with double underscore \_\_) are accessible only within the class.

Protected members (prefixed with single underscore \_) are accessible within the class and its subclasses (convention only, not enforced).

## 12. Types of Polymorphism

Compile-time polymorphism (Method overloading)

Runtime polymorphism (Method overriding)

## 13. Method Overriding and Overloading

Method Overriding: A subclass provides a specific implementation of a method already defined in its superclass.

Method Overloading: Multiple methods with the same name but different parameters (Python doesn't support this directly).

## 14. Operator Overloading

Operator overloading is defining how operators (+, -, \*, /, etc.) behave for custom objects by implementing special methods like \_\_add\_\_, \_\_sub\_\_, etc.

## 15. Magic Functions

Magic functions (or dunder methods) are special methods in Python that start and end with double underscores (e.g., \_\_init\_\_, \_\_str\_\_). They provide functionality to classes.

## 16. Python Module

A Python module is a file containing Python definitions and statements that can be imported and used in other Python programs.

## 17. Creating a Python Module

Create a .py file with Python code

Save it with a meaningful name

Import it in another Python file using import

## 18. Built-in Python Modules

math, os, sys, datetime, random, json, re, collections, itertools, functools

## 19. Circle Module

circle.py:

python

Copy

import math

def area(radius):

return math.pi \* radius \*\* 2

def perimeter(radius):

return 2 \* math.pi \* radius

main.py:

python

Copy

import circle

print(circle.area(5))print(circle.perimeter(5))

## 20. Calculator Module

calculator.py:

python

Copy

def add(a, b):

return a + b

def subtract(a, b):

return a - b

main.py:

python

Copy

import calculator

print(calculator.add(5, 3))print(calculator.subtract(5, 3))

## 21. Module Descriptions

a. sys: Provides access to system-specific parameters and functions.  
b. os: Provides a way of using operating system dependent functionality.  
c. datetime: Supplies classes for manipulating dates and times.  
d. math: Provides mathematical functions and constants.  
e. numpy: Fundamental package for scientific computing with Python.  
f. matplotlib: Comprehensive library for creating static, animated, and interactive visualizations.  
g. pandas: Provides high-performance, easy-to-use data structures and data analysis tools.

## 22. Square Root using math

python

Copy

import math

number = 16print(math.sqrt(number))

## 23. Code Analysis

The code counts and returns the number of lines in the file "abc.txt".

## 24. readline vs readlines()

readline(): Reads a single line from the file.

readlines(): Reads all lines and returns them as a list of strings.

## 25. Closing a File

The close() method is used to close a file in Python. Alternatively, using with statement automatically closes the file.

## 26. Exception Handling Terms

a. try: Block of code to be attempted (might raise an exception).  
b. except: Block of code to execute if there's an exception in the try block.  
c. else: Block of code to execute if the try block does not raise an exception.  
d. finally: Block of code that will be executed no matter what (cleanup actions).

## 27. Opening a File

The open() function is used to open a file in Python. Example: file = open("filename.txt", "r")