1. Basic class and object

```
class Person:
    def greet(self):
        print("Hello!")

p = Person()
p.greet()
```

2. Class with constructor

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

    def greet(self):
        print("Hello,", self.name)

p = Person("Alice")
p.greet()
```

3. Class with multiple objects

```
class Dog:
    def __init__(self, name):
        self.name = name

d1 = Dog("Tommy")
d2 = Dog("Bruno")
print(d1.name)
print(d2.name)
```

4. Class with method using parameters

```
class Calculator:
    def add(self, a, b):
        return a + b

c = Calculator()
print(c.add(5, 3))
```

5. Class with instance variables

```
class Circle:
    def __init__(self, radius):
        self.radius = radius

    def area(self):
        return 3.14 * self.radius ** 2

c = Circle(5)
print(c.area())
```

6. Class with default parameter

```
class Car:
    def __init__(self, model="Toyota"):
        self.model = model

c = Car()
print(c.model)
```

7. Class with user input

```
class Student:
    def __init__(self):
        self.name = input("Enter name: ")

    def show(self):
        print("Name:", self.name)

s = Student()
s.show()
```

8. Class with private attribute

```
class Employee:
    def __init__(self, name):
        self.__name = name

    def show(self):
        print("Name:", self.__name)

e = Employee("John")
e.show()
```

9. Class with multiple methods

```
class Rectangle:
    def __init__(self, l, b):
        self.length = l
        self.breadth = b

    def area(self):
        return self.length * self.breadth

    def perimeter(self):
        return 2 * (self.length + self.breadth)

r = Rectangle(4, 5)
print(r.area())
print(r.perimeter())
```

10. Class with condition

```
class Number:
    def __init__(self, n):
        self.n = n

    def check(self):
        if self.n % 2 == 0:
            print("Even")
        else:
            print("Odd")

num = Number(7)
num.check()
```

11. Class with loop

```
class Table:
    def __init__(self, n):
        self.n = n

    def show(self):
        for i in range(1, 11):
            print(f"{self.n} x {i} = {self.n * i}")
```

```
t = Table(5)
t.show()
```

12. Class with list

```
class Subjects:
    def __init__(self):
        self.subjects = ["Math", "Science", "English"]

    def display(self):
        for sub in self.subjects:
            print(sub)

s = Subjects()
s.display()
```

13. Class with tuple

```
class Coordinates:
    def __init__(self):
        self.point = (10, 20)

    def show(self):
        print("X:", self.point[0], "Y:", self.point[1])

c = Coordinates()
c.show()
```

14. Class with dictionary

```
class Student:
    def __init__(self):
        self.marks = {"Math": 85, "Science": 90}

    def show_marks(self):
        for subject, mark in self.marks.items():
            print(subject, ":", mark)

s = Student()
s.show_marks()
```

15. Class with set

```
class UniqueNumbers:
    def __init__(self):
        self.nums = {1, 2, 3, 2, 1}

    def show(self):
        print(self.nums)

u = UniqueNumbers()
u.show()
```

16. Class comparison

```
class Compare:
    def __init__(self, a):
        self.a = a

    def is_equal(self, other):
        return self.a == other.a

x = Compare(10)
y = Compare(10)
print(x.is_equal(y))
```

19. Class with **str** method

```
class Book:
    def __init__(self, title):
        self.title = title

    def __str__(self):
        return f"Book: {self.title}"

b = Book("Python 101")
print(b)
```

20. Class inheritance simple

```
class Animal:
    def sound(self):
        print("Some sound")

class Dog(Animal):
    def sound(self):
        print("Bark")

d = Dog()
d.sound()
```

```
Bark
```

Start coding or generate with AI.