**PYTHON CODE FOR PRATICE**

**1. Keywords In Python:**

```python

# Sample code 1

def function\_example():

print("This is a function.")

# Sample code 2

print("This is True.")

# Sample code 3

for i in range(3):

print("Loop iteration:", i)

```

**2. Variables In Python:**

```python

# Sample code 1

x = 10

print("Value of x:", x)

# Sample code 2

name = "Alice"

age = 25

print(f"My name is {name} and I am {age} years old.")

# Sample code 3

pi = 3.14159

radius = 5

area = pi \* radius \*\* 2

print("Area of the circle:", area)

```

**3. Getting Input In Python:**

```python

# Sample code 1

name = input("Enter your name: ")

print("Hello, " + name + "!")

# Sample code 2

num = int(input("Enter an integer: "))

print("You entered:", num)

# Sample code 3

height = float(input("Enter your height in meters: "))

weight = float(input("Enter your weight in kilograms: "))

bmi = weight / (height \*\* 2)

print("Your BMI is:", bmi)

```

**5. Single and Multi-line Comment In Python:**

```python

# Sample code 1

# This is a single-line comment

print("Hello, world!")

# Sample code 2

"""

This is a multi-line comment.

It can span multiple lines.

"""

print("Multi-line comments are useful for longer explanations.")

# Sample code 3

# You can use comments to disable code:

# print("This won't be executed.")

```

**6. Type Casting In Python:**

```python

# Sample code 1

x = int("10")

print("x + 5 =", x + 5)

# Sample code 2

y = float("3.14")

print("Value of y:", y)

# Sample code 3

age = int(input("Enter your age: "))

print(f"Next year, you will be {age + 1} years old.")

```

**7. String and String Functions in Python:**

```python

# Sample code 1

message = "Hello, World!"

print("Length of the message:", len(message))

# Sample code 2

name = "Alice"

greeting = f"Hello, {name}!"

print(greeting)

# Sample code 3

sentence = "Python is a versatile language."

print("Is 'Python' in the sentence?", 'Python' in sentence)

```

**8. String Manipulation in Python:**

```python

# Sample code 1

text = "python is fun"

print(text.upper())

# Sample code 2

word = "hello"

repeated\_word = word \* 3

print(repeated\_word)

# Sample code 3

sentence = " Python is easy to learn. "

print(sentence.strip())

```

**9. Arithmetic Operators in Python:**

```python

# Sample code 1

a, b = 10, 3

print("a + b =", a + b)

# Sample code 2

print("a - b =", a - b)

# Sample code 3

print("a \* b =", a \* b)

```

**10. Assignment Operators in Python:**

```python

# Sample code 1

x = 10

x += 5

print(x)

# Sample code 2

y = 8

y -= 2

print(y)

# Sample code 3

z = 3

z \*= 4

print(z)

```

**11. Comparison Operators in Python**:

```python

# Sample code 1

a, b = 5, 10

print("Is a equal to b?", a == b)

# Sample code 2

print("Is a not equal to b?", a != b)

# Sample code 3

print("Is a less than b?", a < b)

```

**12. Logical Operators in Python:**

```python

# Sample code 1

x, y = True, False

print("x and y:", x and y)

# Sample code 2

print("x or y:", x or y)

# Sample code 3

print("not x:", not x)

```

**13. Bitwise Operators in Python**:

```python

# Sample code 1

a, b = 10, 7

print("a & b:", a & b)

# Sample code 2

print("a | b:", a | b)

# Sample code 3

print("a ^ b:", a ^ b)

```

**14. IF Statement in Python:**

```python

# Sample code 1

num = 5

if num > 0:

print("Positive number")

# Sample code 2

age = 18

if age >= 18:

print("You are an adult.")

# Sample code 3

num = 0

if num == 0:

print("Number is zero.")

```

**15. IF - Else Statement in Python:**

```python

# Sample code 1

num = 10

if num % 2 == 0:

print("Even")

else:

print("Odd")

# Sample code 2

x = 7

if x > 0:

print("Positive")

else:

print("Non-positive")

# Sample code 3

num = -3

if num < 0:

print("Negative")

else:

print("Non-negative")

```

**16. ELIF Statement in Python:**

```python

# Sample code 1

score = 85

if score >= 90:

print("A")

elif score >= 80:

print("B")

elif score >= 70:

print("C")

else:

print("D")

# Sample code 2

x = 5

if x > 0:

print("Positive")

elif x == 0:

print("Zero")

else:

print("Negative")

# Sample code 3

num = 25

if num > 100:

print("Greater than 100")

elif num > 50:

print("Between 51 and 100")

else:

print("50 or less")

```

**17. Nested IF Statement in Python:**

```python

# Sample code 1

x = 10

if x > 5:

print("x is greater than 5.")

if x > 8:

print("x is also greater than 8.")

else:

print("x is less than or equal to 8.")

# Sample code 2

age = 18

if age >= 18:

print("You are an adult.")

if age >= 21:

print("You are eligible to drink.")

else:

print("You are not eligible to drink.")

# Sample code 3

num = 10

if num < 0:

print("Negative")

```

**18. While Loop in Python:**

```python

# Sample code 1

count = 0

while count < 5:

print("Count:", count)

count += 1

# Sample code 2

num = 10

while num > 0:

print(num)

num -= 1

# Sample code 3

x = 1

while x <= 100:

if x % 10 == 0:

print(x)

x += 1

```

**19. Continue using While Loop in Python:**

```python

# Sample code 1

num = 0

while num < 5:

num += 1

if num == 3:

continue

print(num)

# Sample code 2

count = 0

while count < 10:

count += 1

if count % 2 == 0:

continue

print("Odd count:", count)

# Sample code 3

x = 0

while x < 10:

x += 1

if x % 3 == 0:

continue

print("Not divisible by 3:", x)

```

**20. Break using While Loop in Python:**

```python

# Sample code 1

num = 1

while True:

print(num)

num += 1

if num == 5:

break

# Sample code 2

count = 0

while count < 10:

count += 1

if count == 5:

break

print("Count:", count)

# Sample code 3

x = 0

while x < 10:

x += 1

if x == 7:

break

print("Value of x:", x)

```

**21. Range in Python**:

```python

# Sample code 1

for i in range(5):

print(i)

# Sample code 2

for j in range(2, 8):

print(j)

# Sample code 3

for k in range(1, 20, 5):

print(k)

```

**22. For Loop in Python:**

```python

# Sample code 1

fruits = ["apple", "banana", "cherry"]

for fruit in fruits:

print(fruit)

# Sample code 2

sentence = "Python is fun!"

for char in sentence:

print(char)

# Sample code 3

numbers = [1, 2, 3, 4, 5]

sum = 0

for num in numbers:

sum += num

print("Sum of numbers:", sum)

```

**23. Nested For Loop in Python:**

```python

# Sample code 1

for i in range(3):

for j in range(2):

print(i, j)

# Sample code 2

rows = 3

columns = 4

for row in range(rows):

for col in range(columns):

print(f"({row}, {col})")

# Sample code 3

size = 5

for i in range(size):

for j in range(i + 1):

print("\*", end="")

print()

```

**24. While Else & For Else:**

```python

# Sample code 1

num = 1

while num <= 5:

print(num)

num += 1

else:

print("Loop finished.")

# Sample code 2

for i in range(5):

if i == 3:

print("Breaking loop at", i)

break

else:

print("Loop finished.")

# Sample code 3

search\_value = 8

numbers = [1, 3, 5, 7, 9]

for num in numbers:

if num == search\_value:

print("Value found:", num)

break

else:

print("Value not found.")

```

**25. Lists in Python:**

```python

# Sample code 1

numbers = [1, 2, 3, 4, 5]

print(numbers)

# Sample code 2

fruits = ["apple", "banana", "cherry"]

print(len(fruits))

# Sample code 3

mixed\_list = [10, "hello", True, 3.14]

print(mixed\_list)

```

**26. Tuple in Python:**

```python

# Sample code 1

my\_tuple = (1, 2, 3)

print(my\_tuple)

# Sample code 2

mixed\_tuple = ("apple", 3, True)

print(mixed\_tuple)

# Sample code 3

nested\_tuple = ((1, 2), ("a", "b"))

print(nested\_tuple)

```

**27. Set in Python:**

```python

# Sample code 1

fruits\_set = {"apple", "banana", "cherry"}

print(fruits\_set)

# Sample code 2

numbers\_set = {1, 2, 3, 4, 5}

print(numbers\_set)

# Sample code 3

mixed\_set = {1, 2, "hello", True}

print(mixed\_set)

```

**28. Dictionary in Python:**

```python

# Sample code 1

student = {"name": "Alice", "age": 20, "major": "Computer Science"}

print(student)

# Sample code 2

grades = {"Math": 90, "Science": 85, "History": 78}

print(grades)

# Sample code 3

person = {

"name": "John",

"age": 25,

"address": {

"street": "123 Main St",

"city": "New York",

"zip": "10001"

}

}

print(person)

```

**29. Identity Operators in Python:**

```python

# Sample code 1

x = 5

y = 5

print(x is y)

# Sample code 2

a = [1, 2, 3]

b = a

print(a is b)

# Sample code 3

c = [1, 2, 3]

print(a is c)

```

**30. Membership operators in Python:**

```python

# Sample code 1

fruits = ["apple", "banana", "cherry"]

print("banana" in fruits)

# Sample code 2

sentence = "Python is fun!"

print("is" not in sentence)

# Sample code 3

numbers = (1, 2, 3, 4, 5)

print(4 in numbers)

```

**31. Functions in Python:**

```python

# Sample code 1

def greet():

print("Hello!")

# Sample code 2

def add\_numbers(a, b):

return a + b

# Sample code 3

def is\_even(num):

return num % 2 == 0

greet()

result = add\_numbers(5, 3)

print("Sum:", result)

print("Is 6 even

?", is\_even(6))

```

**32. Math Functions in Python:**

```python

# Sample code 1

import math

num = 3.75

print("Floor:", math.floor(num))

print("Ceil:", math.ceil(num))

print("Square root:", math.sqrt(num))

# Sample code 2

angle = 45

radians = math.radians(angle)

print("Sin:", math.sin(radians))

print("Cos:", math.cos(radians))

print("Tan:", math.tan(radians))

# Sample code 3

x = 2

y = 3

print("Power:", math.pow(x, y))

print("Absolute:", math.fabs(-10))

```

**33. Lambda Functions in Python:**

```python

# Sample code 1

add = lambda x, y: x + y

print("Result:", add(5, 3))

# Sample code 2

multiply = lambda a, b, c: a \* b \* c

print("Result:", multiply(2, 3, 4))

# Sample code 3

is\_even = lambda num: num % 2 == 0

print("Is 6 even?", is\_even(6))

```

**34. File Handling in Python:**

```python

# Sample code 1 (Writing to a file)

with open("sample.txt", "w") as file:

file.write("Hello, this is a sample text.\n")

file.write("This is the second line.")

# Sample code 2 (Reading from a file)

with open("sample.txt", "r") as file:

content = file.read()

print(content)

# Sample code 3 (Appending to a file)

with open("sample.txt", "a") as file:

file.write("\nThis line is appended to the file.")

```

**35. Exception Handling in Python:**

```python

# Sample code 1

try:

x = 10 / 0

except ZeroDivisionError:

print("Cannot divide by zero.")

# Sample code 2

try:

num = int(input("Enter a number: "))

except ValueError:

print("Invalid input. Please enter a valid number.")

# Sample code 3

try:

my\_list = [1, 2, 3]

print(my\_list[5])

except IndexError:

print("Index out of range.")

```

**36. Reading and Writing CSV Files in Python:**

```python

import csv

# Sample code 1

data = [["Name", "Age", "Country"],

["Alice", 25, "USA"],

["Bob", 30, "Canada"],

["Eve", 22, "UK"]]

with open("data.csv", "w", newline="") as file:

writer = csv.writer(file)

writer.writerows(data)

# Sample code 2

with open("data.csv", "r") as file:

reader = csv.reader(file)

for row in reader:

print(row)

# Sample code 3

with open("data.csv", "r") as file:

reader = csv.DictReader(file)

for row in reader:

print(row["Name"], row["Age"], row["Country"])

```

Of course! Here are 8 additional code examples for the same topics:

**1. Variables In Python:**

```python

# Sample code 4

name = "Bob"

age = 30

city = "New York"

print(f"My name is {name}, I am {age} years old, and I live in {city}.")

# Sample code 5

x = 5

y = 3

z = x + y

print("Sum of x and y:", z)

# Sample code 6

pi = 3.14159

radius = 7

circumference = 2 \* pi \* radius

print("Circumference of the circle:", circumference)

```

**2. Getting Input In Python:**

```python

# Sample code 4

num1 = int(input("Enter the first number: "))

num2 = int(input("Enter the second number: "))

result = num1 + num2

print("Sum of the numbers:", result)

# Sample code 5

name = input("Enter your name: ")

age = input("Enter your age: ")

print(f"Hello, {name}! You are {age} years old.")

# Sample code 6

length = float(input("Enter the length of the rectangle: "))

width = float(input("Enter the width of the rectangle: "))

area = length \* width

print("Area of the rectangle:", area)

```

**3. String and String Functions in Python:**

```python

# Sample code 4

text = "Python is a powerful language."

print(text.lower())

# Sample code 5

sentence = "I love programming in Python."

print(sentence.replace("Python", "Java"))

# Sample code 6

word = "hello"

repeated\_word = word \* 5

print(repeated\_word)

```

**4. String Manipulation in Python:**

```python

# Sample code 4

sentence = " Python is easy to learn. "

print(sentence.strip())

# Sample code 5

text = "Python is fun"

print(text.capitalize())

# Sample code 6

message = "Have a great day!"

print(message.title())

```

**5. Arithmetic Operators in Python:**

```python

# Sample code 4

a, b = 10, 3

print("a / b =", a / b)

# Sample code 5

print("a // b =", a // b)

# Sample code 6

print("a % b =", a % b)

```

**6. Assignment Operators in Python:**

```python

# Sample code 4

x = 10

x \*\*= 3

print(x)

# Sample code 5

y = 8

y //= 2

print(y)

# Sample code 6

z = 3

z %= 2

print(z)

```

**7. Comparison Operators in Python:**

```python

# Sample code 4

a, b = 5, 10

print("Is a greater than b?", a > b)

# Sample code 5

print("Is a less than or equal to b?", a <= b)

# Sample code 6

x = 5

y = 5

print("Are x and y equal?", x == y)

```

**8. Logical Operators in Python:**

```python

# Sample code 4

x, y = True, False

print("x or y:", x or y)

# Sample code 5

a, b = False, False

print("a and b:", a and b)

# Sample code 6

num = 12

print("num < 10 or num > 20:", num < 10 or num > 20)

```

Certainly! Here are 6 additional code examples for the same topics:

**1. IF Statement in Python:**

```python

# Sample code 4

num = int(input("Enter a number: "))

if num % 2 == 0:

print("Even number.")

else:

print("Odd number.")

# Sample code 5

temperature = float(input("Enter the temperature in Celsius: "))

if temperature < 0:

print("Freezing cold!")

elif temperature < 20:

print("Cool weather.")

else:

print("Warm weather.")

# Sample code 6

age = int(input("Enter your age: "))

if age >= 18:

print("You are an adult.")

else:

print("You are a minor.")

```

**2. IF - Else Statement in Python:**

```python

# Sample code 4

num = int(input("Enter a number: "))

if num > 0:

print("Positive number.")

elif num < 0:

print("Negative number.")

else:

print("Zero.")

# Sample code 5

x = 10

y = 20

if x > y:

print("x is greater than y.")

else:

print("x is less than or equal to y.")

# Sample code 6

marks = int(input("Enter your marks: "))

if marks >= 90:

print("A Grade")

elif marks >= 80:

print("B Grade")

elif marks >= 70:

print("C Grade")

else:

print("D Grade")

```

**3. ELIF Statement in Python:**

```python

# Sample code 4

age = int(input("Enter your age: "))

if age < 12:

print("You are a child.")

elif age < 20:

print("You are a teenager.")

elif age < 60:

print("You are an adult.")

else:

print("You are a senior citizen.")

# Sample code 5

num = int(input("Enter a number: "))

if num > 0:

print("Positive number.")

elif num < 0:

print("Negative number.")

else:

print("Zero.")

# Sample code 6

num = int(input("Enter a number: "))

if num > 50:

print("Greater than 50.")

elif num > 25:

print("Greater than 25.")

else:

print("Less than or equal to 25.")

```

**4. Nested IF Statement in Python:**

```python

# Sample code 4

age = int(input("Enter your age: "))

if age >= 18:

print("You are an adult.")

if age >= 21:

print("You are eligible to vote.")

else:

print("You are not eligible to vote.")

else:

print("You are a minor.")

# Sample code 5

x = int(input("Enter a number: "))

if x > 0:

if x % 2 == 0:

print("Positive even number.")

else:

print("Positive odd number.")

else:

print("Non-positive number.")

# Sample code 6

num = int(input("Enter a number: "))

if num >= 0:

if num == 0:

print("Zero.")

else:

print("Positive number.")

else:

print("Negative number.")

```

**5. While Loop in Python:**

```python

# Sample code 4

n = 5

while n > 0:

print(n)

n -= 1

# Sample code 5

num = 1

while num <= 10:

if num % 2 == 0:

print("Even number:", num)

num += 1

# Sample code 6

x = 1

while x <= 100:

if x % 3 == 0 and x % 5 == 0:

print("Multiple of both 3 and 5:", x)

x += 1

```

**6. Continue using While Loop in Python:**

```python

# Sample code 4

num = 0

while num < 10:

num += 1

if num % 2 == 0:

continue

print("Odd number:", num)

# Sample code 5

count = 0

while count < 5:

count += 1

if count == 3:

continue

print("Count:", count)

# Sample code 6

x = 0

while x < 10:

x += 1

if x % 3 == 0:

continue

print("Not divisible by 3:", x)

```

**LIGHT MUCH OF DIFCULT CODE:**

**1. Lists in Python:**

**```python**

**# Sample code 4**

**fruits = ["apple", "banana", "cherry"]**

**print("First fruit:", fruits[0])**

**print("Second fruit:", fruits[1])**

**# Sample code 5**

**numbers = [1, 2, 3, 4, 5]**

**sum\_of\_numbers = sum(numbers)**

**print("Sum of numbers:", sum\_of\_numbers)**

**# Sample code 6**

**names = ["Alice", "Bob", "Charlie"]**

**for name in names:**

**print("Hello, " + name + "!")**

**```**

**2. Tuple in Python:**

**```python**

**# Sample code 4**

**point = (5, 10)**

**print("X coordinate:", point[0])**

**print("Y coordinate:", point[1])**

**# Sample code 5**

**colors = ("red", "green", "blue")**

**for color in colors:**

**print("Color:", color)**

**# Sample code 6**

**empty\_tuple = ()**

**print("Length of an empty tuple:", len(empty\_tuple))**

**```**

**3. Set in Python:**

**```python**

**# Sample code 4**

**fruits\_set = {"apple", "banana", "cherry"}**

**fruits\_set.add("orange")**

**print("Updated set:", fruits\_set)**

**# Sample code 5**

**numbers\_set = {1, 2, 3, 4, 5}**

**squares = {num \*\* 2 for num in numbers\_set}**

**print("Squares:", squares)**

**# Sample code 6**

**languages = {"Python", "Java", "C++"}**

**for lang in languages:**

**print("Language:", lang)**

**```**

**4. Dictionary in Python:**

**```python**

**# Sample code 4**

**student = {"name": "Alice", "age": 20, "major": "Computer Science"}**

**print("Name:", student["name"])**

**print("Age:", student["age"])**

**# Sample code 5**

**grades = {"Math": 90, "Science": 85, "History": 78}**

**total\_marks = sum(grades.values())**

**print("Total marks:", total\_marks)**

**# Sample code 6**

**person = {**

**"name": "John",**

**"age": 25,**

**"address": {**

**"street": "123 Main St",**

**"city": "New York",**

**"zip": "10001"**

**}**

**}**

**print("City:", person["address"]["city"])**

**```**

**5. Identity Operators in Python:**

**```python**

**# Sample code 4**

**x = 5**

**y = 5**

**print("x is y:", x is y)**

**# Sample code 5**

**a = [1, 2, 3]**

**b = a**

**print("a is b:", a is b)**

**# Sample code 6**

**c = [1, 2, 3]**

**print("a is c:", a is c)**

**```**

**6. Membership operators in Python:**

**```python**

**# Sample code 4**

**fruits = ["apple", "banana", "cherry"]**

**print("Is 'orange' in fruits?", 'orange' in fruits)**

**# Sample code 5**

**sentence = "Python is fun!"**

**print("Is 'Java' not in sentence?", 'Java' not in sentence)**

**# Sample code 6**

**numbers = (1, 2, 3, 4, 5)**

**print("Is 4 in numbers?", 4 in numbers)**

**```**

**1. Functions in Python:**

**```python**

**# Sample code 1**

**def greet():**

**print("Hello!")**

**# Sample code 2**

**def add\_numbers(a, b):**

**return a + b**

**# Sample code 3**

**def is\_even(num):**

**return num % 2 == 0**

**# Sample code 4**

**def multiply\_numbers(x, y):**

**return x \* y**

**greet()**

**result = add\_numbers(5, 3)**

**print("Sum:", result)**

**print("Is 6 even?", is\_even(6))**

**print("Product:", multiply\_numbers(2, 4))**

**```**

**2. Date Time Functions in Python:**

**```python**

**# Sample code 1**

**import datetime**

**current\_date = datetime.date.today()**

**print("Current date:", current\_date)**

**# Sample code 2**

**current\_time = datetime.datetime.now()**

**print("Current time:", current\_time)**

**# Sample code 3**

**birthday = datetime.date(2000, 5, 15)**

**print("Year of birth:", birthday.year)**

**print("Month of birth:", birthday.month)**

**# Sample code 4**

**future\_date = current\_date + datetime.timedelta(days=30)**

**print("Date after 30 days:", future\_date)**

**```**

**3. Math Functions in Python:**

**```python**

**# Sample code 1**

**import math**

**print("Square root of 16:", math.sqrt(16))**

**print("Ceiling value of 3.7:", math.ceil(3.7))**

**# Sample code 2**

**print("Absolute value of -5:", abs(-5))**

**print("Factorial of 5:", math.factorial(5))**

**# Sample code 3**

**angles = [0, math.pi/6, math.pi/4, math.pi/3]**

**sines = [math.sin(angle) for angle in angles]**

**print("Sine values:", sines)**

**# Sample code 4**

**base = 2**

**exponent = 5**

**print("2^5 =", math.pow(base, exponent))**

**```**

**4. Try Block in Python:**

**```python**

**# Sample code 1**

**try:**

**x = 10 / 0**

**except ZeroDivisionError:**

**print("Cannot divide by zero.")**

**else:**

**print("Division successful.")**

**# Sample code 2**

**try:**

**num = int(input("Enter a number: "))**

**except ValueError:**

**print("Invalid input. Please enter a valid number.")**

**else:**

**print("You entered:", num)**

**# Sample code 3**

**try:**

**my\_list = [1, 2, 3]**

**print(my\_list[5])**

**except IndexError:**

**print("Index out of range.")**

**else:**

**print("Access successful.")**

**# Sample code 4**

**try:**

**file = open("non\_existent.txt", "r")**

**except FileNotFoundError:**

**print("File not found.")**

**else:**

**print("File opened successfully.")**

**```**

**5. Class and Object in Python:**

**```python**

**# Sample code 1**

**class Car:**

**def \_\_init\_\_(self, make, model):**

**self.make = make**

**self.model = model**

**car1 = Car("Toyota", "Camry")**

**print("Car 1 make:", car1.make)**

**print("Car 1 model:", car1.model)**

**# Sample code 2**

**class Circle:**

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

**self.radius = radius**

**def area(self):**

**return 3.14159 \* self.radius \*\* 2**

**circle1 = Circle(5)**

**print("Area of Circle 1:", circle1.area())**

**# Sample code 3**

**class Student:**

**def \_\_init\_\_(self, name, age):**

**self.name = name**

**self.age = age**

**student1 = Student("Alice", 20)**

**print("Student 1 name:", student1.name)**

**print("Student 1 age:", student1.age)**

**# Sample code 4**

**class Dog:**

**def \_\_init\_\_(self, name, breed):**

**self.name = name**

**self.breed = breed**

**dog1 = Dog("Buddy", "Labrador")**

**print("Dog 1 name:", dog1.name)**

**print("Dog 1 breed:", dog1.breed)**

**```**

**6. Class Attributes in Python:**

**```python**

**# Sample code 1**

**class Circle:**

**pi = 3.14159**

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

**self.radius = radius**

**def area(self):**

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

**circle1 = Circle(5)**

**print("Area of Circle 1:", circle1.area())**

**print("Value of pi:", circle1.pi)**

**# Sample code 2**

**class Rectangle:**

**sides = 4**

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

**self.length = length**

**self.width = width**

**def perimeter(self):**

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

**rectangle1 = Rectangle(3, 4)**

**print("Perimeter of Rectangle 1:", rectangle1.perimeter())**

**print("Number of sides:", rectangle1.sides)**

**# Sample code 3**

**class Person:**

**species = "Homo sapiens"**

**def \_\_init\_\_(self, name, age):**

**self.name = name**

**self.age = age**

**person1 = Person("Alice", 25)**

**print("Person 1 name:", person1.name)**

**print("Person 1 age:", person1.age)**

**print("Species:", person1.species)**

**# Sample code 4**

**class BankAccount:**

**interest\_rate = 0.05**

**def \_\_init\_\_(self, balance):**

**self.balance = balance**

**def calculate\_interest(self):**

**return BankAccount.interest\_rate \* self.balance**

**account1 = BankAccount(1000)**

**print("Interest on account 1:", account1.calculate\_interest())**

**print("Interest rate:", account1.interest\_rate)**

**```**

**7. Instance Attributes in Python:**

**```python**

**# Sample code 1**

**class Car:**

**def \_\_init\_\_(self, make, model):**

**self.make = make**

**self.model = model**

**car1 = Car("Toyota", "Camry")**

**8. Instance Method in Python:**

**```python**

**# Sample code 1**

**class Circle:**

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

**self.radius = radius**

**def area(self):**

**return 3.14159 \* self.radius \*\* 2**

**def circumference(self):**

**return 2 \* 3.14159 \* self.radius**

**circle1 = Circle(5)**

**print("Area of Circle 1:", circle1.area())**

**print("Circumference of Circle 1:", circle1.circumference())**

**# Sample code 2**

**class Student:**

**def \_\_init\_\_(self, name, age):**

**self.name = name**

**self.age = age**

**def display\_info(self):**

**print("Name:", self.name)**

**print("Age:", self.age)**

**student1 = Student("Alice", 20)**

**student1.display\_info()**

**# Sample code 3**

**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)**

**rectangle1 = Rectangle(3, 4)**

**print("Area of Rectangle 1:", rectangle1.area())**

**print("Perimeter of Rectangle 1:", rectangle1.perimeter())**

**# Sample code 4**

**class Dog:**

**def \_\_init\_\_(self, name, breed):**

**self.name = name**

**self.breed = breed**

**def bark(self):**

**print(f"{self.name} says: Woof! Woof!")**

**dog1 = Dog("Buddy", "Labrador")**

**dog1.bark()**

**```**

**9. Init method in Python:**

**```python**

**# Sample code 1**

**class Car:**

**def \_\_init\_\_(self, make, model, year):**

**self.make = make**

**self.model = model**

**self.year = year**

**def display\_info(self):**

**print(f"{self.year} {self.make} {self.model}")**

**car1 = Car("Toyota", "Camry", 2022)**

**car1.display\_info()**

**# Sample code 2**

**class Person:**

**def \_\_init\_\_(self, name, age):**

**self.name = name**

**self.age = age**

**def introduce(self):**

**print(f"Hi, I am {self.name} and I am {self.age} years old.")**

**person1 = Person("Alice", 25)**

**person1.introduce()**

**# Sample code 3**

**class Circle:**

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

**self.radius = radius**

**def area(self):**

**return 3.14159 \* self.radius \*\* 2**

**circle1 = Circle(5)**

**print("Area of Circle 1:", circle1.area())**

**# Sample code 4**

**class Rectangle:**

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

**self.length = length**

**self.width = width**

**def perimeter(self):**

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

**rectangle1 = Rectangle(3, 4)**

**print("Perimeter of Rectangle 1:", rectangle1.perimeter())**

**```**

**10. Class Method in Python:**

**```python**

**# Sample code 1**

**class Circle:**

**pi = 3.14159**

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

**self.radius = radius**

**@classmethod**

**def get\_pi(cls):**

**return cls.pi**

**@classmethod**

**def set\_pi(cls, new\_pi):**

**cls.pi = new\_pi**

**circle1 = Circle(5)**

**print("Value of pi:", circle1.get\_pi())**

**# Sample code 2**

**class Student:**

**total\_students = 0**

**def \_\_init\_\_(self, name):**

**self.name = name**

**Student.total\_students += 1**

**@classmethod**

**def get\_total\_students(cls):**

**return cls.total\_students**

**student1 = Student("Alice")**

**student2 = Student("Bob")**

**print("Total students:", Student.get\_total\_students())**

**# Sample code 3**

**class Rectangle:**

**sides = 4**

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

**self.length = length**

**self.width = width**

**@classmethod**

**def get\_sides(cls):**

**return cls.sides**

**def perimeter(self):**

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

**rectangle1 = Rectangle(3, 4)**

**print("Number of sides:", rectangle1.get\_sides())**

**# Sample code 4**

**class Dog:**

**species = "Canis lupus familiaris"**

**def \_\_init\_\_(self, name):**

**self.name = name**

**@classmethod**

**def get\_species(cls):**

**return cls.species**

**dog1 = Dog("Buddy")**

**print("Species:", dog1.get\_species())**

**```**

**11. Instance Method in Python:**

**```python**

**# Sample code 1**

**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)**

**rectangle1 = Rectangle(4, 6)**

**print("Area of Rectangle:", rectangle1.area())**

**print("Perimeter of Rectangle:", rectangle1.perimeter())**

**# Sample code 2**

**class BankAccount:**

**def \_\_init\_\_(self, balance):**

**self.balance = balance**

**def deposit(self, amount):**

**self.balance += amount**

**def withdraw(self, amount):**

**if self.balance >= amount:**

**self.balance -= amount**

**else:**

**print("Insufficient funds.")**

**def get\_balance(self):**

**return self.balance**

**account1 = BankAccount(1000)**

**print("Initial balance:", account1.get\_balance())**

**account1.deposit(500)**

**print("Balance after deposit:", account1.get\_balance())**

**account1.withdraw(300)**

**print("Balance after withdrawal:", account1.get\_balance())**

**```**

**12. Init method in Python:**

**```python**

**# Sample code 1**

**class Person:**

**def \_\_init\_\_(self, name, age):**

**self.name = name**

**self.age = age**

**def display\_info(self):**

**print(f"Name: {self.name}, Age: {self.age}")**

**person1 = Person("Alice", 25)**

**person1.display\_info()**

**# Sample code 2**

**class Circle:**

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

**self.radius = radius**

**def area(self):**

**return 3.14159 \* self.radius \*\* 2**

**circle1 = Circle(5)**

**print("Area of Circle:", circle1.area())**

**# Sample code 3**

**class Student:**

**def \_\_init\_\_(self, name, age):**

**self.name = name**

**self.age = age**

**def display(self):**

**print(f"Name: {self.name}, Age: {self.age}")**

**student1 = Student("Bob", 22)**

**student1.display()**

**```**

**13. Property Decorator in Python:**

**```python**

**# Sample code 1**

**class Circle:**

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

**self.\_radius = radius**

**@property**

**def radius(self):**

**return self.\_radius**

**@radius.setter**

**def radius(self, value):**

**if value > 0:**

**self.\_radius = value**

**else:**

**print("Invalid radius value.")**

**circle1 = Circle(5)**

**print("Radius:", circle1.radius)**

**circle1.radius = 7**

**print("Updated Radius:", circle1.radius)**

**# Sample code 2**

**class Temperature:**

**def \_\_init\_\_(self, celsius):**

**self.\_celsius = celsius**

**@property**

**def celsius(self):**

**return self.\_celsius**

**@celsius.setter**

**def celsius(self, value):**

**if value < -273.15:**

**print("Temperature cannot be below absolute zero.")**

**else:**

**self.\_celsius = value**

**temp = Temperature(25)**

**print("Temperature in Celsius:", temp.celsius)**

**temp.celsius = -300**

**# Sample code 3**

**class Square:**

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

**self.\_side = side**

**@property**

**def side(self):**

**return self.\_side**

**@side.setter**

**def side(self, value):**

**if value > 0:**

**self.\_side = value**

**else:**

**print("Invalid side length.")**

**square1 = Square(4)**

**print("Side length:", square1.side)**

**square1.side = 5**

**print("Updated side length:", square1.side)**

**```**

**14. Property Decorator Getter Setter in Python:**

**```python**

**# Sample code 1**

**class Circle:**

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

**self.\_radius = radius**

**@property**

**def radius(self):**

**return self.\_radius**

**@radius.setter**

**def radius(self, value):**

**if value > 0:**

**self.\_radius = value**

**else:**

**print("Invalid radius value.")**

**circle1 = Circle(5)**

**print("Radius:", circle1.radius)**

**circle1.radius = 7**

**print("Updated Radius:", circle1.radius)**

**# Sample code 2**

**class Temperature:**

**def \_\_init\_\_(self, celsius):**

**self.\_celsius = celsius**

**@property**

**def celsius(self):**

**return self.\_celsius**

**@celsius.setter**

**def celsius(self, value):**

**if value < -273.15:**

**print("Temperature cannot be below absolute zero.")**

**else:**

**self.\_celsius = value**

**temp = Temperature(25)**

**print("Temperature in Celsius:", temp.celsius)**

**temp.celsius = -300**

**# Sample code 3**

**class Square:**

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

**self.\_side = side**

**@property**

**def side(self):**

**return self.\_side**

**@side.setter**

**def side(self, value):**

**if value > 0:**

**self.\_side = value**

**else:**

**print("Invalid side length.")**

**square1 = Square(4)**

**print("Side length:", square1.side)**

**square1.side = 5**

**print("Updated side length:", square1.side)**

**```**

**15. Property Method in Python:**

**```python**

**# Sample code 1**

**class Circle:**

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

**self.\_radius = radius**

**def get\_radius(self):**

**return self.\_radius**

**def set\_radius(self, value):**

**if value > 0:**

**self.\_radius = value**

**else:**

**print("Invalid radius value.")**

**radius = property(get\_radius, set\_radius)**

**circle1 = Circle(5)**

**print("Radius:", circle1.radius)**

**circle1.radius = 7**

**print("Updated Radius:", circle1.radius)**

**# Sample code 2**

**class Square:**

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

**self.\_side = side**

**def get\_side(self):**

**return self.\_side**

**def set\_side(self, value):**

**if value > 0:**

**self.\_side = value**

**else:**

**print("Invalid side length.")**

**side = property(get\_side, set\_side)**

**square1 = Square(4)**

**print("Side length:", square1.side)**

**square1.side = 5**

**print("Updated side length:", square1.side)**

**# Sample code 3**

**class Temperature:**

**def \_\_init\_\_(self, celsius):**

**self.\_celsius = celsius**

**def get\_celsius(self):**

**return self.\_celsius**

**def set\_celsius(self, value):**

**if value < -273.15:**

**print("Temperature cannot be below absolute zero.")**

**else:**

**self.\_celsius = value**

**celsius = property(get\_celsius, set\_celsius)**

**temp = Temperature(25)**

**print("Temperature in Celsius:", temp.celsius)**

**temp.celsius = -300**

**```**

**16. Class Method Decorator in Python:**

**```python**

**# Sample code 1**

**class Rectangle:**

**sides = 4**

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

**self**

**.length = length**

**self.width = width**

**@classmethod**

**def get\_sides(cls):**

**return cls.sides**

**def area(self):**

**return self.length \* self.width**

**def perimeter(self):**

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

**rectangle1 = Rectangle(3, 4)**

**print("Number of sides:", rectangle1.get\_sides())**

**# Sample code 2**

**class Circle:**

**pi = 3.14159**

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

**self.radius = radius**

**@classmethod**

**def get\_pi(cls):**

**return cls.pi**

**def area(self):**

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

**circle1 = Circle(5)**

**print("Value of pi:", circle1.get\_pi())**

**# Sample code 3**

**class BankAccount:**

**total\_accounts = 0**

**def \_\_init\_\_(self, balance):**

**self.balance = balance**

**BankAccount.total\_accounts += 1**

**@classmethod**

**def get\_total\_accounts(cls):**

**return cls.total\_accounts**

**def deposit(self, amount):**

**self.balance += amount**

**def withdraw(self, amount):**

**if self.balance >= amount:**

**self.balance -= amount**

**else:**

**print("Insufficient funds.")**

**account1 = BankAccount(1000)**

**account2 = BankAccount(2000)**

**print("Total accounts:", BankAccount.get\_total\_accounts())**

**```**

**17. Static Method in Python:**

**```python**

**# Sample code 1**

**class MathUtils:**

**@staticmethod**

**def add(a, b):**

**return a + b**

**@staticmethod**

**def multiply(a, b):**

**return a \* b**

**@staticmethod**

**def square(x):**

**return x \*\* 2**

**sum\_result = MathUtils.add(5, 3)**

**print("Sum:", sum\_result)**

**product\_result = MathUtils.multiply(4, 6)**

**print("Product:", product\_result)**

**square\_result = MathUtils.square(4)**

**print("Square:", square\_result)**

**# Sample code 2**

**class StringUtils:**

**@staticmethod**

**def reverse(text):**

**return text[::-1]**

**@staticmethod**

**def is\_palindrome(text):**

**return text == StringUtils.reverse(text)**

**word = "level"**

**print("Is '{}' a palindrome?".format(word), StringUtils.is\_palindrome(word))**

**# Sample code 3**

**class DateUtils:**

**@staticmethod**

**def is\_leap\_year(year):**

**return (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0)**

**year = 2024**

**print("Is {} a leap year?".format(year), DateUtils.is\_leap\_year(year))**

**```**

**18. Abstraction and Encapsulation in Python:**

**```python**

**# Sample code 1**

**class BankAccount:**

**def \_\_init\_\_(self, balance):**

**self.\_balance = balance**

**def deposit(self, amount):**

**if amount > 0:**

**self.\_balance += amount**

**else:**

**print("Invalid deposit amount.")**

**def withdraw(self, amount):**

**if 0 < amount <= self.\_balance:**

**self.\_balance -= amount**

**else:**

**print("Insufficient funds or invalid withdrawal amount.")**

**def get\_balance(self):**

**return self.\_balance**

**account1 = BankAccount(1000)**

**account1.deposit(500)**

**account1.withdraw(200)**

**print("Balance:", account1.get\_balance())**

**# Sample code 2**

**class Employee:**

**def \_\_init\_\_(self, name, salary):**

**self.\_name = name**

**self.\_salary = salary**

**def get\_name(self):**

**return self.\_name**

**def set\_name(self, name):**

**self.\_name = name**

**def get\_salary(self):**

**return self.\_salary**

**def set\_salary(self, salary):**

**if salary >= 0:**

**self.\_salary = salary**

**else:**

**print("Invalid salary.")**

**employee1 = Employee("Alice", 50000)**

**print("Employee:", employee1.get\_name())**

**print("Salary:", employee1.get\_salary())**

**# Sample code 3**

**class Circle:**

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

**self.\_radius = radius**

**def get\_radius(self):**

**return self.\_radius**

**def set\_radius(self, radius):**

**if radius > 0:**

**self.\_radius = radius**

**else:**

**print("Invalid radius.")**

**def area(self):**

**return 3.14159 \* self.\_radius \*\* 2**

**circle1 = Circle(5)**

**circle1.set\_radius(7)**

**print("Area of Circle:", circle1.area())**

**```**

**19. Single Inheritance in Python:**

**```python**

**# Sample code 1**

**class Animal:**

**def speak(self):**

**pass**

**class Dog(Animal):**

**def speak(self):**

**return "Woof!"**

**class Cat(Animal):**

**def speak(self):**

**return "Meow!"**

**dog = Dog()**

**cat = Cat()**

**print(dog.speak())**

**print(cat.speak())**

**# Sample code 2**

**class Shape:**

**def area(self):**

**pass**

**class Circle(Shape):**

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

**self.radius = radius**

**def area(self):**

**return 3.14159 \* self.radius \*\* 2**

**class Rectangle(Shape):**

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

**self.length = length**

**self.width = width**

**def area(self):**

**return self.length \* self.width**

**circle = Circle(5)**

**rectangle = Rectangle(4, 6)**

**print("Area of Circle:", circle.area())**

**print("Area of Rectangle:", rectangle.area())**

**# Sample code 3**

**class Vehicle:**

**def drive(self):**

**pass**

**class Car(Vehicle):**

**def drive(self):**

**return "Driving a car."**

**class Bicycle(Vehicle):**

**def drive(self):**

**return "Riding a bicycle."**

**car = Car()**

**bicycle = Bicycle()**

**print(car.drive())**

**print(bicycle.drive())**

**```**

**20. Multiple Inheritance in Python:**

**```python**

**# Sample code 1**

**class Dog:**

**def speak(self):**

**return "Woof!"**

**class Cat:**

**def speak(self):**

**return "Meow!"**

**class Pet(Dog, Cat):**

**pass**

**pet = Pet()**

**print(pet.speak())**

**# Sample code 2**

**class Vehicle:**

**def drive(self):**

**pass**

**class Electric:**

**def charge(self):**

**pass**

**class ElectricCar(Vehicle, Electric):**

**def drive(self):**

**return "Driving an electric car."**

**class ElectricBike(Vehicle, Electric):**

**def drive(self):**

**return "Riding an electric bike."**

**electric\_car = ElectricCar()**

**electric\_bike = ElectricBike()**

**print(electric\_car.drive())**

**print(electric\_bike.drive())**

**# Sample code 3**

**class Parent1:**

**def method1(self):**

**return "Method from Parent1"**

**class Parent2:**

**def method2(self):**

**return "Method from Parent2"**

**class Child(Parent1, Parent2):**

**pass**

**child = Child()**

**print(child.method1())**

**print(child.method2())**

**```**

**21. Multilevel Inheritance in Python:**

**```python**

**# Sample code 1**

**class Animal:**

**def speak(self):**

**pass**

**class Dog(Animal):**

**def speak(self):**

**return "Woof!"**

**class Labrador(Dog):**

**def speak(self):**

**return "Lab**

**rador says Woof!"**

**labrador = Labrador()**

**print(labrador.speak())**

**# Sample code 2**

**class Vehicle:**

**def drive(self):**

**pass**

**class Car(Vehicle):**

**def drive(self):**

**return "Driving a car."**

**class ElectricCar(Car):**

**def drive(self):**

**return "Driving an electric car."**

**electric\_car = ElectricCar()**

**print(electric\_car.drive())**

**# Sample code 3**

**class Grandparent:**

**def method1(self):**

**return "Method from Grandparent"**

**class Parent(Grandparent):**

**def method2(self):**

**return "Method from Parent"**

**class Child(Parent):**

**pass**

**child = Child()**

**print(child.method1())**

**print(child.method2())**

**```**

**22. Function Overriding in Python:**

**```python**

**# Sample code 1**

**class Shape:**

**def area(self):**

**pass**

**class Circle(Shape):**

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

**self.radius = radius**

**def area(self):**

**return 3.14159 \* self.radius \*\* 2**

**class Rectangle(Shape):**

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

**self.length = length**

**self.width = width**

**def area(self):**

**return self.length \* self.width**

**circle = Circle(5)**

**rectangle = Rectangle(4, 6)**

**print("Area of Circle:", circle.area())**

**print("Area of Rectangle:", rectangle.area())**

**# Sample code 2**

**class Parent:**

**def speak(self):**

**return "Hello, I am the parent."**

**class Child(Parent):**

**def speak(self):**

**return "Hi, I am the child."**

**parent = Parent()**

**child = Child()**

**print(parent.speak())**

**print(child.speak())**

**# Sample code 3**

**class Animal:**

**def speak(self):**

**pass**

**class Dog(Animal):**

**def speak(self):**

**return "Woof!"**

**class Cat(Animal):**

**def speak(self):**

**return "Meow!"**

**dog = Dog()**

**cat = Cat()**

**print(dog.speak())**

**print(cat.speak())**

**```**

**23. Handling Diamond Problem in Python:**

**```python**

**# Sample code 1**

**class A:**

**def speak(self):**

**return "A speaks."**

**class B(A):**

**def speak(self):**

**return "B speaks."**

**class C(A):**

**def speak(self):**

**return "C speaks."**

**class D(B, C):**

**pass**

**d = D()**

**print(d.speak())**

**# Sample code 2**

**class A:**

**def method(self):**

**return "Method of A"**

**class B(A):**

**pass**

**class C(A):**

**def method(self):**

**return "Method of C"**

**class D(B, C):**

**pass**

**d = D()**

**print(d.method())**

**# Sample code 3**

**class A:**

**def speak(self):**

**return "A speaks."**

**class B(A):**

**pass**

**class C(A):**

**def speak(self):**

**return "C speaks."**

**class D(B, C):**

**pass**

**d = D()**

**print(d.speak())**

**```**

**24. Operator Overloading in Python:**

**```python**

**# Sample code 1**

**class Vector:**

**def \_\_init\_\_(self, x, y):**

**self.x = x**

**self.y = y**

**def \_\_add\_\_(self, other):**

**return Vector(self.x + other.x, self.y + other.y)**

**def \_\_sub\_\_(self, other):**

**return Vector(self.x - other.x, self.y - other.y)**

**def \_\_str\_\_(self):**

**return f"({self.x}, {self.y})"**

**v1 = Vector(1, 2)**

**v2 = Vector(3, 4)**

**v3 = v1 + v2**

**v4 = v2 - v1**

**print("v1 + v2 =", v3)**

**print("v2 - v1 =", v4)**

**# Sample code 2**

**class ComplexNumber:**

**def \_\_init\_\_(self, real, imaginary):**

**self.real = real**

**self.imaginary = imaginary**

**def \_\_add\_\_(self, other):**

**real = self.real + other.real**

**imaginary = self.imaginary + other.imaginary**

**return ComplexNumber(real, imaginary)**

**def \_\_sub\_\_(self, other):**

**real = self.real - other.real**

**imaginary = self.imaginary - other.imaginary**

**return ComplexNumber(real, imaginary)**

**def \_\_str\_\_(self):**

**return f"{self.real} + {self.imaginary}i"**

**c1 = ComplexNumber(2, 3)**

**c2 = ComplexNumber(4, 5)**

**c3 = c1 + c2**

**c4 = c2 - c1**

**print("c1 + c2 =", c3)**

**print("c2 - c1 =", c4)**

**# Sample code 3**

**class Point:**

**def \_\_init\_\_(self, x, y):**

**self.x = x**

**self.y = y**

**def \_\_eq\_\_(self, other):**

**return self.x == other.x and self.y == other.y**

**def \_\_lt\_\_(self, other):**

**return self.x < other.x and self.y < other.y**

**p1 = Point(2, 3)**

**p2 = Point(4, 5)**

**p3 = Point(2, 3)**

**print("p1 == p2:", p1 == p2)**

**print("p1 == p3:", p1 == p3)**

**print("p1 < p2:", p1 < p2)**

**```**

**25. Abstract Base Class in Python:**

**```python**

**# Sample code 1**

**from abc import ABC, abstractmethod**

**class Shape(ABC):**

**@abstractmethod**

**def area(self):**

**pass**

**class Circle(Shape):**

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

**self.radius = radius**

**def area(self):**

**return 3.14159 \* self.radius \*\* 2**

**class Rectangle(Shape):**

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

**self.length = length**

**self.width = width**

**def area(self):**

**return self.length \* self.width**

**circle = Circle(5)**

**rectangle = Rectangle(4, 6)**

**print("Area of Circle:", circle.area())**

**print("Area of Rectangle:", rectangle.area())**

**# Sample code 2**

**from abc import ABC, abstractmethod**

**class Vehicle(ABC):**

**@abstractmethod**

**def drive(self):**

**pass**

**class Car(Vehicle):**

**def drive(self):**

**return "Driving a car."**

**class Bicycle(Vehicle):**

**def drive(self):**

**return "Riding a bicycle."**

**car = Car()**

**bicycle = Bicycle()**

**print(car.drive())**

**print(bicycle.drive())**

**# Sample code 3**

**from abc import ABC, abstractmethod**

**class Animal(ABC):**

**@abstractmethod**

**def speak(self):**

**pass**

**class Dog(Animal):**

**def speak(self):**

**return "Woof!"**

**class Cat(Animal):**

**def speak(self):**

**return "Meow!"**

**dog = Dog()**

**cat = Cat()**

**print(dog.speak())**

**print(cat.speak())**

**```**

**26. Open a File in Python:**

**```python**

**# Sample code 1**

**try:**

**file = open("sample.txt", "r")**

**content = file.read()**

**print(content)**

**file.close()**

**except FileNotFoundError:**

**print("File not found.")**

**except Exception as e:**

**print("An error occurred:", str(e))**

**# Sample code 2**

**try:**

**file = open("data.txt", "r")**

**lines = file.readlines()**

**for line in lines:**

**print(line.strip())**

**file.close()**

**except FileNotFoundError:**

**print("File not found.")**

**except Exception as e:**

**print("An error occurred:", str(e))**

**# Sample code 3**

**try:**

**with open("numbers.txt", "r") as file:**

**numbers = file.readlines()**

**total = sum(int(num.strip()) for num in numbers)**

**print("Sum of numbers:", total)**

**except FileNotFoundError:**

**print("File not found.")**

**except Exception as e:**

**print("An error occurred:", str(e))**

**```**

**27. Delete a File in Python:**

**```python**

**# Sample code 1**

**import os**

**try:**

**os.remove("sample.txt")**

**print("File deleted successfully.")**

**except FileNotFoundError:**

**print("File not found.")**

**except Exception as e:**

**print("An error occurred:", str(e))**

**# Sample code 2**

**import os**

**try:**

**os.remove("data.txt")**

**print("File deleted successfully.")**

**except FileNotFoundError:**

**print("File not found.")**

**except Exception as e:**

**print("An error occurred:", str(e))**

**# Sample code 3**

**import os**

**try:**

**os.remove("numbers.txt")**

**print("File deleted successfully.")**

**except FileNotFoundError:**

**print("File not found.")**

**except Exception as e:**

**print("An error occurred:", str(e))**

**```**