Python Programming

Python Classes/Objects

```
# Defining a class
class Person:
   def __init__(self, name, age):
      self.name = name
      self.age = age

   def __str__(self):
      return f"{self.name}({self.age})"

   def myfunc(self):  # An object method
      print(f"Hello, my name is {self.name}")


# Creating an object
p1 = Person("John", 36)

# Accessing attributes and methods
print(p1)
p1.myfunc()
```

```
John(36)
Hello, my name is John
```

Inheritance and Polymorphism

```python
# Parent Class
class Person:
    def __init__(self, first_name, last_name):
        self.first_name = first_name
        self.last_name = last_name

    def full_name(self):
        return f"{self.first_name} {self.last_name}"

    def greet(self):
        return f"Hello, my name is {self.full_name()}."

# Child Class
class Student(Person):
    def __init__(self, first_name, last_name, student_id):
        super().__init__(first_name, last_name)
        self.student_id = student_id

    def student_info(self):
        return f"ID: {self.student_id}, Name: {self.full_name()}"

# Another Child Class
class Teacher(Person):
    def __init__(self, first_name, last_name, subject):
        super().__init__(first_name, last_name)
        self.subject = subject

    def teach(self):
        return f"{self.full_name()} is teaching {self.subject}."

# Instantiate Parent and Child Classes
person = Person("Jane", "Smith")
print(person.greet())

student = Student("John", "Doe", "S12345")
print(student.greet())
print(student.student_info()

teacher = Teacher("Emily", "Davis", "Mathematics")
print(teacher.greet())
print(teacher.teach())
```

```
Hello, my name is Jane Smith.
Hello, my name is John Doe.
ID: S12345, Name: John Doe
Hello, my name is Emily Davis.
Emily Davis is teaching Mathematics.
```

Iterators

```
class MyNumbers:
    def __iter__(self):
        self.a = 1
        return self

    def __next__(self):
        if self.a <= 5:
            x = self.a
            self.a += 1
            return x
        else:
            raise StopIteration

my_numbers = MyNumbers()
my_iter = iter(my_numbers)

for num in my_iter:
    print(num)
```

```
1
2
3
4
5
```

Date/Time

```
import datetime

# Getting the current date and time
current_datetime = datetime.datetime.now()
print("Current date and time:", current_datetime)

# Extracting specific parts of the date/time
print("Year:", current_datetime.year)
print("Month:", current_datetime.month)
print("Day:", current_datetime.day)
print("Hour:", current_datetime.hour)
print("Minute:", current_datetime.minute)
print("Second:", current_datetime.second)

# Creating a specific date
specific_date = datetime.datetime(2023, 12, 25)  # Year, Month, Day
print("Specific date:", specific_date)

# Formatting the current date/time into a readable string
formatted_date = current_datetime.strftime("%Y-%m-%d %H:%M:%S")
print("Formatted date and time:", formatted_date)

# Parsing a string into a datetime object
date_string = "2024-12-28 15:45:10"
parsed_date = datetime.datetime.strptime(date_string, "%Y-%m-%d %H:%M:%S")
print("Parsed date:", parsed_date)

# Using the date and time classes
current_date = datetime.date.today()
print("Current date:", current_date)

specific_time = datetime.time(14, 30, 45)  # Hour, Minute, Second
print("Specific time:", specific_time)

# Calculating the difference between two dates
date1 = datetime.datetime(2024, 1, 1)
date2 = datetime.datetime(2023, 1, 1)
difference = date1 - date2
print("Difference in days:", difference.days)
```

Current date and time: 2024-12-28 15:59:54.884711
Year: 2024
Month: 12
Day: 28
Hour: 15
Minute: 59
Second: 54
Specific date: 2023-12-25 00:00:00
Formatted date and time: 2024-12-28 15:59:54
Parsed date: 2024-12-28 15:45:10
Current date: 2024-12-28
Specific time: 14:30:45
Difference in days: 365

Python Math

```
import math

# Constants
print("Value of pi:", math.pi)  # Value of pi
print("Value of e:", math.e)    # Euler's number

# Mathematical functions

# Absolute value
x = -5
print("Absolute value of -5:", math.fabs(x))

# Rounding values
print("Round 5.67 to nearest integer:", round(5.67))
print("Round 5.67 down:", math.floor(5.67))  # Round down
print("Round 5.67 up:", math.ceil(5.67))    # Round up

# Power and square root
print("2 raised to the power of 3:", math.pow(2, 3))
print("Square root of 16:", math.sqrt(16))

# Trigonometric functions
# Converting degrees to radians
angle_degrees = 90
angle_radians = math.radians(angle_degrees)
print(f"90 degrees in radians:", angle_radians)

# Sine, cosine, and tangent
print("Sine of 90 degrees:", math.sin(math.radians(90)))  # Sin of 90 degrees
print("Cosine of 0 degrees:", math.cos(math.radians(0)))  # Cos of 0 degrees
print("Tangent of 45 degrees:", math.tan(math.radians(45)))  # Tan of 45
degrees

# Inverse trigonometric functions
print("Inverse sine of 1:", math.asin(1))  # Inverse sine
print("Inverse cosine of 0:", math.acos(0))  # Inverse cosine

# Logarithmic functions
print("Natural log of e:", math.log(math.e))   # Natural logarithm (log base e)
print("Log base 10 of 100:", math.log10(100))  # Logarithm with base 10

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

```
Value of pi: 3.141592653589793
Value of e: 2.718281828459045
Absolute value of -5: 5.0
Round 5.67 to nearest integer: 6
Round 5.67 down: 5
Round 5.67 up: 6
2 raised to the power of 3: 8.0
Square root of 16: 4.0
90 degrees in radians: 1.5707963267948966
Sine of 90 degrees: 1.0
Cosine of 0 degrees: 1.0
Tangent of 45 degrees: 0.9999999999999999
Inverse sine of 1: 1.5707963267948966
Inverse cosine of 0: 1.5707963267948966
Natural log of e: 1.0
Log base 10 of 100: 2.0
Factorial of 5: 120
Greatest common divisor of 36 and 60: 12
Hyperbolic sine of 1: 1.1752011936438014
Hyperbolic cosine of 1: 1.5430806348152437
Combinations of 5 items taken 3 at a time (C(5, 3)): 10
Permutations of 5 items taken 3 at a time (P(5, 3)): 60
```

```
# Greatest common divisor
a = 36
b = 60
print("Greatest common divisor of 36 and 60:", math.gcd(a, b))  # GCD of 36 and
60

# Hyperbolic functions
print("Hyperbolic sine of 1:", math.sinh(1))
print("Hyperbolic cosine of 1:", math.cosh(1))

# Combinations and permutations
n = 5
k = 3
print(f"Combinations of 5 items taken 3 at a time (C(5, 3)):", math.comb(n, k))
# Combinations
print(f"Permutations of 5 items taken 3 at a time (P(5, 3)):", math.perm(n, k))
# Permutations
```

User Input

```
# Getting user input using input() function
name = input("Enter your name: ")
print(f"Hello, {name}!")

# Getting numerical input and converting it to an integer
age = int(input("Enter your age: "))
print(f"You are {age} years old.")

# Getting numerical input and converting it to a float
height = float(input("Enter your height in meters: "))
print(f"Your height is {height} meters.")

# Handling multiple inputs in one line
x, y = input("Enter two numbers separated by a space: ").split()
x = int(x)
y = int(y)
print(f"You entered {x} and {y}. The sum is {x + y}.")

# Using input to interact with the program more dynamically
operation = input("Do you want to add or subtract? (Enter 'add' or 'subtract'):
").strip().lower()

if operation == "add":
   a = int(input("Enter the first number: "))
   b = int(input("Enter the second number: "))
   print(f"The sum is {a + b}.")
elif operation == "subtract":
   a = int(input("Enter the first number: "))
   b = int(input("Enter the second number: "))
   print(f"The difference is {a - b}.")
else:
   print("Invalid operation.")
```

```
Enter your name: Carter
Hello, Carter!
Enter your age: 35
You are 35 years old.
Enter your height in meters: 1.7
Your height is 1.7 meters.
Enter two numbers separated by a space: 1 2
You entered 1 and 2. The sum is 3.
Do you want to add or subtract? (Enter 'add' or
'subtract'): add
Enter the first number: 2
Enter the second number: 3
The sum is 5.
```

String Formatting

```
# Basic String Formatting with f-strings (Python 3.6+)
name = "Alice"
age = 30
greeting = f"Hello, my name is {name} and I am {age} years old."
print(greeting)

# String Formatting using the format() method
greeting2 = "Hello, my name is {} and I am {} years old.".format(name, age)
print(greeting2)

# Using positional and keyword arguments in the format() method
greeting3 = "Hello, my name is {0} and I am {1} years old. {0} is learning
Python.".format(name, age)
print(greeting3)

# Using keyword arguments in the format() method
greeting4 = "Hello, my name is {name} and I am {age} years
old.".format(name=name, age=age)
print(greeting4)

# Formatting numbers (e.g., decimals, integers)
pi = 3.141592653589793
formatted_pi = f"The value of pi is approximately {pi:.2f}"
print(formatted_pi)

# Padding and aligning strings with format()
text = "Hello"
aligned_text = f"{text:>10}"  # Right-align the text
print(aligned_text)

aligned_text2 = f"{text:<10}"  # Left-align the text
print(aligned_text2)

aligned_text3 = f"{text:^10}"  # Centre-align the text
print(aligned_text3)

# Padding numbers with leading zeros
number = 42
formatted_number = f"{number:05}"
print(formatted_number)

# Using format() with numbers
number = 1234567
formatted_number2 = "{:,}".format(number)
```

```
Hello, my name is Alice and I am 30 years old.
Hello, my name is Alice and I am 30 years old.
Hello, my name is Alice and I am 30 years old. Alice is
learning Python.
Hello, my name is Alice and I am 30 years old.
The value of pi is approximately 3.14
     Hello
Hello
Hello
00042
1,234,567
45.00%
The result of 5 + 10 is 15.
```

```
print(formatted_number2)

# Formatting percentages
value = 0.45
formatted_percentage = f"{value:.2%}"  # 45.00%
print(formatted_percentage)

# Combining string formatting with expressions
x = 5
y = 10
result = f"The result of {x} + {y} is {x + y}."
print(result)
```