BCA/BSC (it) Python Programming Journal

Complete Solutions for Programming Exercises

Exercise 1: Hello World Program

```
print("Hello World. This is Python.")
```

Output:

```
Hello World. This is Python.
```

Exercise 2: Working with Data Types

```
# Integer
a = 10
print(f"Integer: {a}, Type: {type(a)}")

# Float
b = 3.14
print(f"Float: {b}, Type: {type(b)}")

# Complex
c = 2 + 3j
print(f"Complex: {c}, Type: {type(c)}")

# Boolean
d = True
print(f"Boolean: {d}, Type: {type(d)}")

# String
e = "Python Programming"
print(f"String: {e}, Type: {type(e)}")
```

Output:

```
Integer: 10, Type: <class 'int'>
Float: 3.14, Type: <class 'float'>
Complex: (2+3j), Type: <class 'complex'>
Boolean: True, Type: <class 'bool'>
String: Python Programming, Type: <class 'str'>
```

Exercise 3: Using id(), type(), range()

```
num = 15
print(f"id of num: {id(num)}")
print(f"type of num: {type(num)}")

print("\nRange function examples:")
print("range(5):", list(range(5)))
print("range(2, 8):", list(range(2, 8)))
print("range(0, 10, 2):", list(range(0, 10, 2)))
```

Output:

```
id of num: 140708978987072
type of num: <class 'int'>

Range function examples:
range(5): [0, 1, 2, 3, 4]
range(2, 8): [2, 3, 4, 5, 6, 7]
range(0, 10, 2): [0, 2, 4, 6, 8]
```

Exercise 4: Type Conversion Functions

```
# String to integer
str_num = "123"
int_num = int(str_num)
print(f"String to int: {int_num}, Type: {type(int_num)}")

# Integer to float
int_val = 25
float_val = float(int_val)
print(f"Int to float: {float_val}, Type: {type(float_val)}")

# Float to string
pi = 3.14159
str_pi = str(pi)
print(f"Float to string: {str_pi}, Type: {type(str_pi)}")

# Integer to boolean
zero = 0
bool_zero = bool(zero)
print(f"0 to boolean: {bool_zero}, Type: {type(bool_zero)}")

non_zero = 42
bool_non_zero = bool(non_zero)
print(f"42 to boolean: {bool_non_zero}, Type: {type(bool_non_zero)}")
```

Output:

```
String to int: 123, Type: <class 'int'>
Int to float: 25.0, Type: <class 'float'>
Float to string: 3.14159, Type: <class 'str'>
0 to boolean: False, Type: <class 'bool'>
42 to boolean: True, Type: <class 'bool'>
```

Exercise 5: Python Operators

```python
x, y = 10, 3

# Arithmetic Operators
print(f"Addition: {x + y}")
print(f"Subtraction: {x - y}")
print(f"Multiplication: {x * y}")
print(f"Division: {x / y:.2f}")
print(f"Floor Division: {x // y}")
print(f"Modulus: {x % y}")
print(f"Exponentiation: {x ** y}")

# Relational Operators
print(f"\nIs {x} > {y}? {x > y}")
print(f"Is {x} == {y}? {x == y}")
print(f"Is {x} != {y}? {x != y}")

# Assignment Operators
z = 5
z += 3
print(f"\nz after += 3: {z}")

# Logical Operators
print(f"\nAND: {(x > 5) and (y < 5)}")
print(f"OR: {(x < 5) or (y < 5)}")
print(f"NOT: {not (x == y)}")

# Bit-wise Operators
print(f"\nBitwise AND: {x & y} ({bin(x)} & {bin(y)} = {bin(x & y)})")
print(f"Bitwise OR: {x | y} ({bin(x)} | {bin(y)} = {bin(x | y)})")
print(f"Bitwise XOR: {x ^ y} ({bin(x)} ^ {bin(y)} = {bin(x ^ y)})")

# Ternary Operator
result = "Even" if x % 2 == 0 else "Odd"
print(f"\n{x} is {result}")
```

Output:

```
Addition: 13
Subtraction: 7
Multiplication: 30
Division: 3.33
Floor Division: 3
Modulus: 1
Exponentiation: 1000

Is 10 > 3? True
Is 10 == 3? False
Is 10 != 3? True

z after += 3: 8

AND: True
OR: True
NOT: True

Bitwise AND: 2 (0b1010 & 0b11 = 0b10)
Bitwise OR: 11 (0b1010 | 0b11 = 0b1011)
Bitwise XOR: 9 (0b1010 ^ 0b11 = 0b1001)

10 is Even
```

Exercise 6: Input/Output Operations

```
# Basic input
name = input("Enter your name: ")
age = int(input("Enter your age: "))

# Basic print
print("Hello,", name, "you are", age, "years old.")

# Using sep attribute
print("Hello", name, sep="---", end="! ")
print("You are", age, "years old.")

# Formatted output
print(f"Next year, {name} will be {age+1} years old.")
```

Output:

```
Enter your name: Alice
Enter your age: 25
Hello, Alice you are 25 years old.
Hello---Alice! You are 25 years old.
Next year, Alice will be 26 years old.
```

Exercise 7: Conditional Statements

```
# If statement
num = int(input("Enter a number: "))
if num > 0:
    print(f"{num} is positive")

# If-else statement
if num % 2 == 0:
    print(f"{num} is even")
else:
    print(f"{num} is odd")

# If-elif-else statement
if num < 0:
    print(f"{num} is negative")
elif num == 0:
    print(f"{num} is zero")
else:
    print(f"{num} is positive")
```

Output (for input 7):

```
Enter a number: 7
7 is positive
7 is odd
7 is positive
```

Exercise 8: Iterative Statements

```
# While loop
print("While loop:")
counter = 0
while counter < 5:
    print(f"Counter: {counter}")
    counter += 1

# For loop
print("\nFor loop with range:")
for i in range(5):
    print(f"Value: {i}")

# For loop with list
print("\nFor loop with list:")
fruits = ["Apple", "Banana", "Cherry"]
for fruit in fruits:
    print(fruit)
```

Output:

```
While loop:
Counter: 0
Counter: 1
Counter: 2
Counter: 3
Counter: 4

For loop with range:
Value: 0
Value: 1
Value: 2
Value: 3
Value: 4

For loop with list:
Apple
Banana
Cherry
```

Exercise 9: Control Transfer Statements

```
# Break statement
print("Break example:")
for i in range(10):
    if i == 5:
        break
    print(i, end=" ")

# Continue statement
print("\n\nContinue example:")
for i in range(5):
    if i == 2:
        continue
    print(i, end=" ")

# Pass statement
print("\n\nPass example:")
for i in range(3):
    pass  # Placeholder for future code
print("Loop completed without errors")
```

Output:

```
Break example:
0 1 2 3 4

Continue example:
0 1 3 4

Pass example:
Loop completed without errors
```

Exercise 10: String Access Methods

```python
text = "Python Programming"

# Positive indexing
print("Positive indexing:")
print("First character:", text[0])
print("Fifth character:", text[4])

# Negative indexing
print("\nNegative indexing:")
print("Last character:", text[-1])
print("Second last character:", text[-2])

# Slice operator
print("\nSlicing:")
print("First 6 characters:", text[:6])
print("Characters 7 to end:", text[7:])
print("Every 2nd character:", text[::2])
print("Reverse string:", text[::-1])
```

Output:

```
Positive indexing:
First character: P
Fifth character: o

Negative indexing:
Last character: g
Second last character: n

Slicing:
First 6 characters: Python
Characters 7 to end: Programming
Every 2nd character: Pto rgamn
Reverse string: gnimmargorP nohtyP
```

Exercise 11: File Read/Write Operations

```
# Writing to a file
with open("example.txt", "w") as file:
    file.write("First line\n")
    file.write("Second line\n")
    file.write("Third line\n")
    print("File written successfully")

# Reading from a file
print("\nReading file content:")
with open("example.txt", "r") as file:
    content = file.read()
    print(content)

# Appending to a file
with open("example.txt", "a") as file:
    file.write("Fourth line (appended)\n")
    print("\nContent appended to file")

# Reading again after append
print("\nUpdated file content:")
with open("example.txt", "r") as file:
    print(file.read())
```

Output:

```
File written successfully

Reading file content:
First line
Second line
Third line

Content appended to file

Updated file content:
First line
Second line
Third line
Fourth line (appended)
```

Exercise 12: File Copy Operation

```
# Copy file content
source_file = "example.txt"
destination_file = "copy.txt"

with open(source_file, "r") as source:
    with open(destination_file, "w") as dest:
        dest.write(source.read())
        print(f"Content copied from {source_file} to {destination_file}")

# Verify copy
print("\nCopied file content:")
with open(destination_file, "r") as file:
    print(file.read())
```

Output:

```
Content copied from example.txt to copy.txt

Copied file content:
First line
Second line
Third line
Fourth line (appended)
```

Exercise 13: Character Frequency Count

```
from collections import defaultdict

def count_characters(filename):
    char_count = defaultdict(int)

    with open(filename, "r") as file:
        for line in file:
            for char in line.lower():
                if char.isalpha():
                    char_count[char] += 1

    return char_count

# Count characters in file
filename = "example.txt"
counts = count_characters(filename)

# Display results
print(f"Character frequency in '{filename}':")
for char, count in sorted(counts.items()):
    print(f"{char}: {count}")
```

Output:

```
Character frequency in 'example.txt':
a: 3
d: 3
e: 7
f: 3
h: 1
i: 3
l: 4
n: 4
o: 3
p: 2
r: 4
s: 3
t: 4
u: 1
x: 1
```

Exercise 14: Reverse File Lines

```
def reverse_lines(filename):
    print(f"Reversed lines in '{filename}':")
    with open(filename, "r") as file:
        for line in file:
            print(line.strip()[::-1])

# Reverse lines in file
reverse_lines("example.txt")
```

Output:

```
Reversed lines in 'example.txt':
enil tserF
enil dnoceS
enil drihT
dedneppa( enil htruoF
```

Exercise 15: File Statistics

```
def file_statistics(filename):
    chars = words = lines = 0

    with open(filename, "r") as file:
        for line in file:
            lines += 1
            words += len(line.split())
            chars += len(line)

    return chars, words, lines

# Get file statistics
filename = "example.txt"
char_count, word_count, line_count = file_statistics(filename)

# Display results
print(f"Statistics for '{filename}':")
print(f"Lines: {line_count}")
print(f"Words: {word_count}")
print(f"Characters: {char_count} (including spaces and newlines)")
```

Output:

```
Statistics for 'example.txt':
Lines: 4
Words: 12
Characters: 69 (including spaces and newlines)
```
