



COMP 1023 Introduction to Python Programming

Exercises on Functions

COMP 1023 Teaching Team

Department of Computer Science & Engineering
The Hong Kong University of Science and Technology, Hong Kong SAR, China



Exercise 1

A hexagonal number is defined as $n(2n - 1)$ for $n = 1, 2, \dots$, and so on. So, the first few numbers are 1, 6, 15, 28, Write a function with the following header that returns a hexagonal number:

```
def get_hexagonal_number(n):
```

For example, `get_hexagonal_number(1)` returns 1 and `get_hexagonal_number(2)` returns 6. Write a test program that uses this function to display the first 50 hexagonal numbers with 5 numbers on each line.

Exercise 1 Solutions

```
def get_hexagonal_number(n):  
    """Return the nth hexagonal number using the formula  $n(2n - 1)$ """  
    return n * (2 * n - 1)  
  
# Test program to display first 50 hexagonal numbers  
def main():  
    print("The first 50 hexagonal numbers:")  
  
    for i in range(1, 51):  
        # Calculate hexagonal number  
        hex_num = get_hexagonal_number(i)  
        # Print the number with formatting  
        print(f"{hex_num:8}", end="")  
        # Print new line after every 5 numbers  
        if i % 5 == 0:  
            print()  
  
# Call the test program  
if __name__ == "__main__": main()
```

Exercise 2

Write a function that computes the product of the digits in an integer. Use the following function header:

```
def product_of_digits(n):
```

For example, `product_of_digits(234)` returns 24 ($2 * 3 * 4$).

Hint: Use the `%` operator to extract digits, and the `//` operator to remove the extracted digit. For instance, to extract 4 from 234, use `234 % 10 (= 4)`. To remove 4 from 234, use `234 // 10 (= 23)`. Use a loop to repeatedly extract and remove the digits until all the digits are extracted.

Write a test program that prompts the user to enter a positive integer and displays the product of all its digits.

Enter an integer: 12345

The product of digits for 12345 is 120

Exercise 2 Solutions

```
def product_of_digits(n):  
    """Compute the product of all digits in an integer"""  
    if num == 0: # Handle special case of 0  
        return 0  
    product = 1  
    while num > 0: # Extract digits until no digits left  
        digit = num % 10 # Extract the last digit  
        product *= digit # Multiply the digit to product  
        num = num // 10 # Remove the last digit  
    return product  
  
def main(): # Test program  
    number = int(input("Enter an integer: ")) # Prompt user for input  
    result = product_of_digits(number) # Calculate product of digits  
    # Display the result  
    print(f"The product of digits for {number} is {result}")  
  
# Call the test program  
if __name__ == "__main__": main()
```

Exercise 3

Write a function to display a pattern as follows:

```
1
2 1
3 2 1
4 3 2 1
...
```

The function header is

```
def display_pattern(n):
```

Write a test program that prompts the user to enter an integer `n` and invokes `display_pattern(n)` to display the pattern.

Enter line number: 5

```
1
2 1
3 2 1
4 3 2 1
5 4 3 2 1
```

Exercise 3 Solutions

```
def display_pattern(n):  
    """Display a pattern where each line counts down from the line number to 1"""  
    for i in range(1, n + 1):  
        # Print numbers from i down to 1  
        for j in range(i, 0, -1):  
            print(j, end=" ")  
        # Move to the next line after each row  
        print()  
  
    # Test program  
def main():  
    # Prompt user for input  
    n = int(input("Enter line number: "))  
  
    # Display the pattern  
    display_pattern(n)  
  
    # Call the test program  
if __name__ == "__main__": main()
```

Exercise 4

Create a Python program named `my_triangle.py` that contains the following two functions:

```
# Return true if the sum of any two sides is greater than the third side  
def is_valid(side1, side2, side3):
```

```
# Return the area of the triangle  
def area(side1, side2, side3):
```

Write a test program that reads three sides for a triangle and computes the area if the input is valid. Otherwise, it displays that the input is invalid. The formula for computing the area of a triangle is as follows:

$$s = (side1 + side2 + side3)/2$$
$$area = \sqrt{s(s - side1)(s - side2)(s - side3)}$$

```
Enter edge1: 1
```

```
Enter edge2: 1.1
```

```
Enter edge3: 1
```

```
The area of the triangle is 0.45934055993347683
```

Exercise 4 Solutions

```
import math
def is_valid(side1, side2, side3):
    """Return True if the sum of any two sides is greater than the third side"""
    return (side1 + side2 > side3) and (side1 + side3 > side2) and (side2 + side3 > side1)

def area(side1, side2, side3):
    """Return the area of the triangle using the formula"""
    s = (side1 + side2 + side3) / 2
    triangle_area = math.sqrt(s * (s - side1) * (s - side2) * (s - side3))
    return triangle_area

def main():
    # Read three sides from user
    side1 = float(input("Enter edge1: "))
    side2 = float(input("Enter edge2: "))
    side3 = float(input("Enter edge3: "))

    if is_valid(side1, side2, side3):
        # Check if the sides form a valid triangle
        triangle_area = area(side1, side2, side3) # Calculate and display the area
        print(f"The area of the triangle is {triangle_area}")
    else:
        print("The input is invalid - these sides cannot form a triangle")

# Run the test program
if __name__ == "__main__": main()
```

Exercise 5

Write a function that converts milliseconds to hours, minutes, and seconds using the following header:

```
def convert_millis(millis):
```

The function returns a string as hours:minutes:seconds. For example, `convert_millis(5500)` returns the string `0:0:5`, `convert_millis(100000)` returns the string `0:1:40`, and `convert_millis(555550000)` returns the string `154:19:10`. Write a test program that prompts the user to enter a value for milliseconds and displays a string in the format of hours:minutes:seconds.

```
Enter time in milliseconds: 1123241637  
312:0:41
```

Exercise 5 Solutions

```
def convert_millis(millis):  
    """Convert milliseconds to hours, minutes, and seconds format"""  
    total_seconds = millis // 1000 # Convert milliseconds to total seconds  
  
    # Calculate hours, minutes, and seconds  
    hours = total_seconds // 3600  
    remaining_seconds = total_seconds % 3600  
    minutes = remaining_seconds // 60  
    seconds = remaining_seconds % 60  
  
    return f"{hours}:{minutes}:{seconds}" # Return formatted string  
  
def main(): # Test program  
    milliseconds = int(input("Enter time in milliseconds: ")) # Prompt user for input  
    result = convert_millis(milliseconds) # Convert and display the result  
    print(result)  
  
# Call the test program  
if __name__ == "__main__": main()
```

Exercise 6

Write a program with function that returns the area of a hexagon. Prompt the user to enter the side of hexagon. Area of hexagon is $(3\sqrt{3}s^2)/2$.

```
def area(s):
```

Exercise 6 Solutions

```
import math

def area(s):
    """Return the area of a regular hexagon with side length s"""
    return (3 * math.sqrt(3) * s ** 2) / 2

def main():
    # Prompt user to enter the side length
    side = float(input("Enter the side of the hexagon: "))

    # Calculate the area
    hexagon_area = area(side)

    # Display the result
    print(f"The area of the hexagon with side {side} is {hexagon_area}")

# Call the main function
if __name__ == "__main__":
    main()
```