```python
# Invoice Generation

def generate_invoice(customer_name, product_details):
    """Generates a simple invoice.

    Args:
        customer_name (str): The name of the customer.
        product_details (list of dict): A list where each dictionary
                                        represents a product with
'name' and 'price' keys.
    """
    print("----- INVOICE -----")
    print(f"Customer: {customer_name}\n")
    print("Product\t\tPrice")
    print("-----------------------")
    total_price = 0
    for item in product_details:
        print(f"{item['name']}\t\t${item['price']:.2f}")
        total_price += item['price']
    print("-----------------------")
    print(f"Total:\t\t${total_price:.2f}")
    print("-----------------------")

# Example usage:
customer = "Alice Smith"
products = [
    {"name": "Laptop", "price": 1200.50},
    {"name": "Mouse", "price": 25.00},
    {"name": "Keyboard", "price": 75.75}
]
generate_invoice(customer, products)
```

```
----- INVOICE -----
Customer: Alice Smith

Product          Price
-----------------------
Laptop           $1200.50
Mouse      $25.00
Keyboard         $75.75
-----------------------
Total:           $1301.25
-----------------------
```

```python
# 2 - Student Report

def generate_student_report(student_name, grade):
    """Prints a formatted student report.

    Args:
```

```
        student_name (str): The name of the student.
        grade (str): The final grade of the student.
    """
    print(f'"Student: {student_name}\nFinal Grade: {grade}"')

# Example usage:
student = "Arjun Mehta"
final_grade = "A+"
generate_student_report(student, final_grade)
```

```
"Student: Arjun Mehta
Final Grade: A+"
```

```
# 3 Remove Extra Spaces

def remove_extra_spaces(message):
    """Removes extra spaces from a user-entered message.

    Args:
        message (str): The input message string.

    Returns:
        str: The message with extra spaces removed.
    """
    return " ".join(message.split())

# Example usage:
user_input = "This   is  a  message with   extra   spaces."
cleaned_message = remove_extra_spaces(user_input)
print(f"Original message: '{user_input}'")
print(f"Cleaned message: '{cleaned_message}'")
```

```
Original message: 'This   is  a  message with   extra   spaces.'
Cleaned message: 'This is a message with extra spaces.'
```

```
# 4. Count "good" (Case-Insensitive)

def count_good(feedback_string):
    """Counts the number of times "good" appears in a string (case-
insensitive).

    Args:
        feedback_string (str): The input feedback string.

    Returns:
        int: The number of times "good" appears.
    """
    normalized_string = feedback_string.lower()
    count = normalized_string.count("good")
    return count
```

```python
# Example usage:
feedback = "This product is Good and the service is also good. It's
really GOOD!"
good_count = count_good(feedback)
print(f"The word 'good' appears {good_count} times.")
```

```
The word 'good' appears 3 times.
```

```python
# 5 Password Check

def check_password(password):
    """Checks if a password meets the following criteria:
    - At least 1 uppercase letter
    - At least 1 lowercase letter
    - At least 1 digit
    - Is at least 8 characters long

    Args:
        password (str): The password to check.

    Returns:
        bool: True if the password meets all criteria, False
otherwise.
    """
    has_upper = False
    has_lower = False
    has_digit = False

    if len(password) < 8:
        return False

    for char in password:
        if char.isupper():
            has_upper = True
        elif char.islower():
            has_lower = True
        elif char.isdigit():
            has_digit = True

    return has_upper and has_lower and has_digit

# Example usage:
password_to_check1 = "P@sswOrd123"
password_to_check2 = "weak"
print(f"'{password_to_check1}' is valid:
{check_password(password_to_check1)}")
print(f"'{password_to_check2}' is valid:
{check_password(password_to_check2)}")
```

```
'P@sswOrd123' is valid: True
'weak' is valid: False
```