Table of Contents

[Objective: 1](#_Toc170076914)

[Tools and Libraries Used: 1](#_Toc170076915)

[Script Descriptions: 1](#_Toc170076916)

[User Information and Validation: 1](#_Toc170076917)

[Even Number Identification: 1](#_Toc170076918)

[Temperature Conversion: 1](#_Toc170076919)

[Minimum and Maximum Numbers: 2](#_Toc170076920)

[Students Results in a Python Dictionary: 2](#_Toc170076921)

[Inventory Management: 2](#_Toc170076922)

[Learning and Insights: 2](#_Toc170076923)

[Conclusion: 3](#_Toc170076924)

Report on Python Script Development

# Objective:

The objective of this project was to develop a set of Python scripts that perform various tasks related to user input, data validation, mathematical operations, and inventory management. The scripts were designed to cover a range of fundamental concepts in Python programming while providing practical solutions to common problems.

# Tools and Libraries Used:

* Python 3: The core programming language used for script development.
* re module: Used for regular expression-based validation of email addresses.
* scipy.constants module: Utilized for temperature conversion operations.

# Script Descriptions:

## User Information and Validation:

This script collects user information such as name, age, email, and favorite number.

It validates the email format using regular expressions.

If the email format is valid, it displays a message with the collected user information; otherwise, it prompts the user to enter a valid email address.

## Even Number Identification:

The script determines whether a given integer is even or odd.

It defines a function Is\_even(number) that takes an integer input from the user and prints whether it's even or odd using the modulus operator.

## Temperature Conversion:

Two approaches for temperature conversion are implemented in this script.

* Approach 1: Defines conversion functions for Celsius to Fahrenheit and Fahrenheit to Celsius.
* Approach 2: Utilizes the scipy.constants module, specifically the convert\_temperature function, for temperature conversion.

Both approaches prompt the user for temperature and scale inputs, perform the conversion, and display the result.

## Minimum and Maximum Numbers:

This script prompts the user to enter five numbers and identifies the maximum and minimum numbers from the entered list using the max() and min() functions.

It then prints the maximum and minimum numbers.

## Students Results in a Python Dictionary:

The script collects details of three students including their names, ages, and grades.

It validates the grades to ensure they are within a predefined set of valid grades.

The student details are stored in a dictionary with the student name as the key and a tuple containing age and grade as the value.

Finally, it displays the student details.

## Inventory Management:

This script manages an inventory system by allowing the user to update the quantities of items in the inventory.

It initializes an inventory dictionary with some predefined items and quantities.

Then, it prompts the user to update the quantities of three items by adding or removing quantities.

It ensures that the quantity of any item doesn't go below zero and displays the updated inventory.

# Learning and Insights:

* Regular Expressions: The project involved the use of regular expressions for email validation, showcasing their utility in data validation tasks.
* Function Definitions: The scripts extensively used function definitions to modularize code and promote reusability.
* Input Validation: Input validation was emphasized in various scripts to ensure data integrity and prevent runtime errors.
* External Libraries: Utilizing external libraries like scipy.constants demonstrated how pre-built functions can streamline complex operations.
* Documentation and Resources: Various online resources, including Python documentation and tutorial websites, were referenced to understand concepts and implementation details.

# Conclusion:

The development of these Python scripts provided valuable hands-on experience in applying fundamental programming concepts to practical problems. The project enhanced proficiency in areas such as data validation, function definition, and interaction with external libraries. Moving forward, further exploration of Python's ecosystem and continued practice will contribute to advancing skills and tackling more complex programming challenges.