HyperionDev

OOP – Classes

Model Answer Approach

Visit our website

Auto-graded task

This Python program simulates an email application using object-oriented programming (OOP) principles and functions. The core of the program revolves around the `Email` class, which encapsulates the attributes of an email such as the sender, subject line, content, and read status. Each email object can be marked as read using the `mark_as_read()` method, which toggles the `has_been_read` attribute from `False` to `True`.

The program utilises functions like `populate_inbox()` to initialise the `inbox` with sample emails and `list_emails()` to display all email subjects with corresponding indexes for selection. Users can choose to read specific emails, mark them as read if unread, or view only unread emails. This modular approach enhances code maintainability, scalability, and reusability by separating functionalities into different functions that perform specific tasks. Additionally, this approach demonstrates how methods can be used to update the value stored within a given attribute for a specific object.

Common pitfalls include not properly creating objects when populating the `inbox`. Failing to create objects correctly makes it challenging to work with each email stored within the `inbox`. Another potential issue is not ensuring that the selected email index is within a valid range before displaying the email and updating the `has_been_read` attribute. These mistakes can lead to errors and unexpected behaviour in the program.