

How Well Your Solution Meets the Requirements of the Scenario

The program meets all the requirements of the scenario well.

The member of staff who is using the program will be able to input the customer's details including their name, address and phone number.

After inputting the customer's details, the member of staff can then input the number of parcels to collect. Once the user has stated how many parcels they need collecting, the program will prompt the user for information about each package. Information collected on each package includes the height, length, width and weight of the package, it also includes if the user wants to add a signature on delivery or if the user wants to add parcel tracking. This will loop until the requested number of parcels have been inputted.

Once the user has inputted all of their parcels the program will display the customer's details. Additionally it will display the price of each individual package as well as the added cost of each additional features (signature and tracking) and if any additional features were added it will also display the total cost of that package. Once it has displayed the cost of all packages individually it will display the total cost of all collection, it will then inform the user to 'Press any key to close!' so they have a chance to read all of the data.

The Quality and Performance of Your Program

For all string inputs, if the input is left blank the user will receive the message 'Invalid input.' and will be asked to provide the value again, this ensures all data is entered, although it does not prevent the user from incorrect spellings, typing mistakes or other incorrect inputs.

For all integer inputs if the user inputs a non-integer value the user will be given the error 'Invalid input. Not an integer' and will be asked to provide the value again. Additionally if the user enters a number that is not between the minimum and maximum allowed the user will receive a different error, in this case of the number of parcels the error would be 'Invalid input. Not between 1 and 6'.

For all Boolean inputs if the user presses a key other than Y or N the input will be ignored. There may be some confusion at first as Boolean inputs do not require an enter press or display errors for invalid keypresses, however it allows the user to input data faster which would be more beneficial to a business and an experienced user.

The customer's details and parcel details are stored as structs to provide a good framework for scaling the program and providing additional features such as storing the data of multiple customers. Additionally the usage of provides a clearer connection between the data than other approaches such as parallel arrays, thus making the program more easily maintainable.

The phone number is currently stored as a string, this allows users to include symbols such as brackets or '+' which may appear in some numbers. Whilst this is useful behaviour it would be beneficial to perform additional checks with regular expressions (Regex) to ensure that a phone number is in a valid format.

The minimum and maximum parcels are stored as constants to make maintaining and editing the program easier as changing the constant value will update it throughout the program. Similar checks are done whenever the user is prompted for an input.

The program was thoroughly tested and passed all tests to ensure that is a high quality program. Due to this being done under examination conditions there are some tests that were unable to be

conducted. These tests were not included in the test log. The tests would involve testing on other operating systems, at minimum this would include MacOS, Linux (Multiple of the most commonly used distributions) and Windows 11. As .NET framework is not specifically listed as supporting MacOS and Linux it may need re-compiling or rewriting to function on these systems. Linux is especially important as many machines in industry are running on a Linux distribution although this is less common among end devices that users will interact with. Additionally, these tests would ideally be automated via unit testing to simplify the process for future developers.

The Choices You Made About Coding Conventions

The coding conventions used are the standard coding conventions used in C# as stated by Microsoft. These conventions were chosen as they are well known by C# developers of all levels which should make maintenance, scaling, and improvements easier.

The program is a console based application running on .NET Framework in C#. It was decided to be a console based program as it will be able to run on the majority of systems due to being less resource intensive. Additionally, by using a console instead of a GUI (graphical user interface) the program is easier to maintain as it is less complex.

The program was also decided to be developed in a more procedural approach. While this is less common in C# which heavily supports object oriented programming the scenario does not require objects. This can also affect performance as object oriented programs add further layers of abstraction which can lead to less efficient programs.

The code has been commented to aid in maintenance and future development. All methods include an XML comment which details the methods purpose, parameters and return values. By using XML comments documentation can be easily produced and Visual Studio 2022's intellisense can provide information about methods in pop-ups. Furthermore, methods have been grouped into related regions which can help future developers find what they are looking for easily.

The Changes You Made During the Development Process

During the development process there were few changes made.

The majority of changes were the result of some tests that had initially failed due to incorrectly written logic, these were made during the process of testing. As the code had changed during testing all tests that were ran with previous versions of the code were repeated to ensure the program was still fully functional.

Some other changes that were made during development was refactoring the code to split repeated logic out into functions, this made the main logical flow of the program simpler to read which improves readability and maintainability.

The pseudocode and flowchart do not fully represent the implementation of the program as some concepts such as for loops, structs and arrays are hard to convey in these formats whilst staying easily understandable, however they represent a logical method for solving the problem in a mostly similar manner.