2.3 PRODUCING ROBUST PROGRAMS

DEFENSIVE DESIGN

Programmers try to protect their programs by testing them to reduce the number of errors, predicting how users might misuse their program and trying to prevent it and making sure their code is well maintained.

Input Sanitisation – removes any unwanted characters that have been entered into a program

Input Validation – Checks if the data meets certain criteria before passing it through the program. The following validation checks can be used:

Presence check	Checks that data has been entered
Length check	Checks the data is the correct length
Range check	Checks the data is within a set range
Format check	Checks it's in the correct format (Eg:dd/mm/yy)
Check digit	Checks numerical data is entered correctly
Look-up table	Checks against a table of accepted values

Authentication – Where a program confirms the identity of a user before giving them access to the full program. This could be done through usernames and passwords.

Maintainability – Code that has been well maintained is easy to edit without causing errors. A well maintained code will have **comments** to help other programmers understand the code, as well as **appropriate names for variables and sub programs**, and **indentation** so that it is easy for programmers to see the flow of the program. Global variables should only be used where necessary so that they don't impact on the rest of your code.

TESTING

A program should be tested to check for any errors.

Final Testing – The program goes is tested once at the end of development. Everything is tested in one go.

Iterative testing - a program is tested and then changes are made as it goes through the development cycle again. It may go through this process a few times to make sure it is exactly what the customer wants.

Test data can fit into 3 different categories:

Normal	Data which is likely to be entered into the program and should be accepted
Extreme/ boundary	Data on the limit of what should be accepted
Erroneous	Data that should not be accepted

TYPES OF ERROR

A program should be tested to check for any errors.

Syntax Error – something which doesn't fit the rules or grammar of the programming language.

Logic Error – the program runs but not as expected. Eg: < user instead of >.