2.3 Producing Robust Programs

Keywords & Definitions

Authentication: The process of determining the identity of a user

Input sanitisation: removing any unwanted characters before passing data through the program

Input validation: checking if data meets certain criteria before passing it into the program

Testing: take measures to check the quality, performance, or reliability

Maintain: uphold the program to ensure it runs efficiently

Defensive Design

Computer programs should be designed to ensure that they can cope with unexpected or erroneous input from users

Programmers try to protect their programs through defensive design, they will try to:

- Anticipate how users might misuse their program, then attempt to prevent it from happening
- Ensure their code is well-maintained
- Reduce the number of errors in the code through testing

Validation Checks

Check digit	the last one or two digits in a code are used to check the other digits are correct	bar code readers in supermarkets use check digits
Format check	checks the data is in the right format	a National Insurance number is in the form LL 99 99 99 L where L is any letter and 9 is any number
Length check	checks the data isn't too short or too long	a password which needs to be six letters long
Lookup table	looks up acceptable values in a table	there are only seven possible days of the week
Presence check	checks that data has been entered into a field	in most databases a key field cannot be left blank
Range check	checks that a value falls within the specified range	number of hours worked must be less than 50 and more than 0
Spell check	looks up words in a dictionary	MS Word uses red lines to underline misspelt words

Validation: checking if data meets certain criteria before passing it into the program

Input Sanitisation

- Cleaning up the data that is inputted
- Data sanitisation trims or strips strings, removing unwanted characters from strings
- For example, Dave not dav%e, the % would be removed

This ensures that the input is correct and contains only the permitted characters, letters and symbols

Authentication

- Confirming the identity of a user, usually through the use of passwords.
- Common ways to increase security:
  - force users to use strong passwords and change them regularly
- limit the number of failed attempts before locking

2.2 Programming Techniques

Maintainability

Your program should be easy to maintain - programmers should make sure that it is well-maintained

Comments (#): useful for explaining what the key features of a program do

Indentation: used to separate different statements in a program - clearly see the flow

Variables/sub programs: should be name so that they refer to what they actually are

Testing

Testing is just as important as the programming itself!

Three main types of errors that can occur:

Runtime errors	Syntax errors	Logic errors
• Errors which may cause program errors or the computer to crash even if there appears to be nothing wrong with the program code. • They are only detected once the program is executed • Examples could be: Running out of memory	• mistakes in the way that the code is written. • Translators can only execute a program if it is syntactically correct. • Common syntax errors include: • spelling mistakes • incorrect use of punctuation • use of capital letters	• a logic error is a bug in a program that causes it to operate incorrectly, but not to terminate or crash. • Examples could be: • Incorrectly using logical operators, eg expecting a program to stop when the value of a variable reaches 5, but using <5 instead of <=5

Iterative testing is testing the code as you create it - line by line OR a section at a time

Final or Terminal testing is carried out at the end of the program when it has been written.

Testing is often completed in a test plan which sets out:

- **The test number**
- **The data entered**
- **The type of test data**
- **The expected outcome**
- **The result of the test**
- **Action required as a result of the test**

Test Data:

Normal/Valid data – data that is correct

In range data – max and min values at the limit of what could entered

Invalid – data that the program should not accept (wrong data type for example)

Null Value – when no data is entered or left blank to test what happens

World of work links

Programmer, IT Technician, Software Engineer, Teacher, Systems Architecture, Data Engineer, Software Developer