Defensive Design

Key Words

authentication	Verifying the identity of a user.
biometrics	Authentication techniques that rely on measurable physical characteristics (eg fingerprint or iris) that can be automatically checked.
comments	Used by a programmer to explains sections of code.
defensive design	Designing a program that runs correctly and continues to run no matter what actions a user takes.
indentation	Indenting makes it easy to see where structures begin and end. Conditions and iterations should be indented. Code inside procedures and functions should be indented.
input	Data which is inserted into a system for processing and/or storage.
input sanitisation	Checks data that is entered and removes anything that might be potentially dangerous.
invalid data	Data that a program cannot process and should not accept
maintainability	The process of ensuring that a program is easy to understand, modify and update.
valid data	Sensible, possible data that a program should accept and be able to process.
validation	Checking input data is sensible and in the right format.

Considerations

- The purpose of defensive design is to ensure that a program runs correctly and continues to run no matter what actions a user takes.
- Thinking about potential problems before starting the development of a new program will eliminate many of the problems that could occur later in the process and improve the reliability of the solution.

input validation

Validation is the process of checking data when it is entered to see it conforms to a rule.

Validation does not ensure that the data entered is correct, just that it is possible and sensible.

Data can be validated to check:

Type	The input is in the correct data type. E.g. Integer, Real, String	E.g: Enter Age (Integer)
Range	The input is within a correct range. E.g. Between 1 and 2	E.g: Input Grade (0-9)
Presence	Some data has been entered. E.g. Reject blank inputs	E.g: Nothing typed (" ")
Format	The input is in the correct format. E.g. dd/mm/yyyy	E.g: Date of Birth (01/01/2005)
Length	The input has the correct number of characters. E.g. 8 or more chars	E.g: Enter password (Minimum 8 characters)

```
number = 0
while number < 1 or number > 10:
    number = int(input("enter a number between 1 and 10"))
print("You entered ", number)
```

input sanitisation

Checks data that is entered and removes anything that might be potentially dangerous. A good example of this is on a website form. A hacker might try to gain access to a website's data through a SQL injection attack.

authentication

Authentication is the process of a user confirming that they are who they say they are on a computer system.

Username & Password	The most common way of doing this is to ask for a username and password. The entered details are then checked against a database containing valid accounts.
Two-factor authentication	This means that after the user enters a valid user name and password, the system sends an SMS text 'authentication code' to the designated mobile phone. They then have to enter this as well.
Biometric authentication	This method checks some physical feature of the authorised person such as their fingerprint. The user puts their thumb on a fingerprint entry device, the data is sent off to a database and checked against their valid data.

Maintainability

The purpose of maintainability is to ensure that, over time, a program can be easily maintained to either add a new feature or to fix a problem.

The understanding of the program, how it works and the purpose of the code will be made easier if the program is written in a maintainable style. This includes using:

- **sub programs**
  - Self contained section of code that performs a specific task
  - Makes the program structure really clear to help with future maintenance
- **comments**
  - Comments are lines in programs that provide information about what the different parts of the program do.
- **sensible variable names**
  - Programmers should choose a variable name that reflects the purpose of the variable and the data it is intended to hold.

`H` - poor choice. What does it mean?

`height` - better choice. The value to be inputted will be a number.

- **indentation**
  - Code within selections or iterations should be indented. This allows the programmer to easily see which code falls within the selection or iteration, and where it ends.

```
do
    for count = 1 to 10
        print("Hey!")
    next count
    again = input("Enter y to go again")
    again.lower
until again !="y"
```

Quizizz | **Extended Reading** | **Exam Questions** | **Video** | **Revision Techniques** | **Quizlet** | **Specification**

BOHUNT EDUCATION TRUST