



F6.2 Authentication

Robust programs · GCSE · OCR J277 2.3.1, AQA 8525 3.2.11, Edexcel 1CP2
6.4.4 · about 15 min

What this lesson is about

Username and passwords, limiting attempts, and stronger ways to prove who someone is.

- Something you know
- Say less when it fails
- Limiting attempts
- Better than a password
- Authentication and validation together

Questions

5 marks in all

1. What is authentication?

[1 mark]

- ☐ A Checking a user is who they claim to be
- ☐ B Checking data is sensible
- ☐ C Encrypting a message
- ☐ D Testing a program

Answer: A. Authentication checks identity, often with a username and password.

2. What does this program print?

[1 mark]

```
operators = {"ada": "robot42", "alan": "turing1"}  
name, password = "ada", "turing1"  
print(name in operators and operators[name] == password)
```

Answer:

False

turing1 is a real password, but it belongs to alan, not ada.

3. Why should a login say "access denied" rather than "no such user"?

[1 mark]

- ☐ A It does not tell an attacker which usernames exist
- ☐ B It is shorter
- ☐ C Python requires it
- ☐ D It stops typing errors

Answer: A. Revealing valid usernames lets an attacker focus on guessing their passwords.

4. Why limit the number of login attempts?

[1 mark]

- ☐ A To stop an attacker guessing passwords over and over
- ☐ B To save memory
- ☐ C To make the program shorter
- ☐ D Because users forget passwords

Answer: A. Unlimited tries make guessing, by a person or a program, possible.

5. A password and a code sent to your phone is an example of what?

[1 mark]

- ☐ A Two-factor authentication
- ☐ B Validation
- ☐ C Biometrics
- ☐ D A range check

Answer: A. Two different kinds of proof: something you know and something you have.

The task: operator login

Use `operators = {"ada": "robot42", "alan": "turing1"}`. Ask `Username?` and `Password?` up to three times. For each wrong pair print `access denied`. When a pair is right, print `welcome <name>`, turn the LED green and drive 20 cm forward. The task types `alan` and `wrong`, then `alan` and `turing1`.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

operators = {"ada": "robot42", "alan": "turing1"}
```

The hint students can ask for: Give the loop a fixed number of attempts. Each time, check the name exists and that the stored password matches it before letting them in, and stop asking once they are.

A solution

```
from bugbot import *
connect()
operators = {"ada": "robot42", "alan": "turing1"}
user = None
for attempt in range(3):
    name = input("Username? ")
    password = input("Password? ")
    if name in operators and operators[name] == password:
        user = name
        break
    print("access denied")
if user:
    print("welcome", user)
    led("green")
    forward(50, distance=20)
```

Any program that meets the task's checks is marked correct in the simulator; this is one way, not the only way.