



F6.1 Defensive design and validation

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

Name _____

Class _____

Date _____

What this lesson is about

Anticipating misuse; presence, type, range, length, format and look-up checks.

- What goes wrong without it
- The validation checks
- Checking in order
- Asking until it is valid
- Length, format and look-up
- Anticipating misuse

Questions

6 marks in all

1. Which validation check makes sure something was entered at all? [1 mark]

- ☐ A Presence check
- ☐ B Range check
- ☐ C Length check
- ☐ D Format check

2. A distance must be from 1 to 50. Which check is that? [1 mark]

- ☐ A Range check
- ☐ B Presence check
- ☐ C Look-up check
- ☐ D Type check

3. What does this program print? [1 mark]

```
for answer in ["30", "", "lots", "-5"]:  
    print(answer.isdigit())
```

-
4. Why check the type of an answer before its range? [1 mark]

- ☐ A Converting text that is not a number to int would crash
- ☐ B Range checks are slower
- ☐ C Type checks are required first by Python
- ☐ D It makes no difference

5. A user types 30 when they meant 20, and the range is 1 to 50. What does validation do? [1 mark]

- ☐ A Accepts it: validation checks data is sensible, not correct
- ☐ B Rejects it
- ☐ C Changes it to 20
- ☐ D Asks them to type it twice

6. Checking a colour is one of red, green or blue is which check?

[1 mark]

- ☐ A Look-up check
- ☐ B Length check
- ☐ C Range check
- ☐ D Presence check

The task: a validated drive

Write a program that asks `How far, 1 to 50 cm?` until the answer is valid, then drives that far forward. For an empty answer print `you must type something`; for one that is not a whole number print `that is not a whole number`; for a number out of range print `it must be between 1 and 50`. The task types an empty answer, then `fifty`, then `500`, then `30`.

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

far = int(input("How far, 1 to 50 cm? "))
forward(60, distance=far)
```

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/f6-1-defensive-design-and-validation/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Accept distances with a decimal point, such as `12.5`, while still rejecting `twelve`.
2. Add a check that refuses to drive if `distance()` shows a wall closer than the distance asked for, plus 5 cm.
3. Validate a class code: exactly six characters, only capital letters and digits.