



F12.4 Copyright and licences

Technology and society · GCSE · OCR J277 1.6.1, AQA 8525 3.8, Edexcel
1CP2 5.2.3 · about 15 min

What this lesson is about

Proprietary and open source software, common licences, and Creative Commons.

- Proprietary software
- Open source software
- Reading a licence
- Doing it properly

Questions

5 marks in all

1. What does open source mean? [1 mark]

- ☐ A The source code is published and may be read, changed and shared
- ☐ B The software is free of charge only
- ☐ C The software has no licence
- ☐ D The software is out of date

Answer: A. Open source is about the licence and the code, not only the price.

2. Give an advantage of proprietary software. [1 mark]

- ☐ A It is usually supported and tested by the company
- ☐ B You can change the source code
- ☐ C It is always free
- ☐ D You can share it freely

Answer: A. Support and documentation are usual reasons to pay.

3. What does a GPL licence require if you distribute your changes? [1 mark]

- ☐ A You must release your version under the GPL too
- ☐ B You must pay the author
- ☐ C You must keep the changes secret
- ☐ D You must apply for a patent

Answer: A. This is called copyleft.

4. What does CC BY mean for an image? [1 mark]

- ☐ A You may use it if you credit the creator
- ☐ B You may not use it
- ☐ C You may use it only for schools
- ☐ D You own it

Answer: A. BY stands for attribution.

5. Which is a disadvantage of open source for a school?

[1 mark]

- ☐ A Support may come from volunteers rather than a company
- ☐ B It cannot be changed
- ☐ C It is always expensive
- ☐ D It has no source code

Answer: A. Paid support is available for some projects, but it is not guaranteed.

The task: may I use it?

Write `allowed(licence, use)` that returns True or False for a proposed use, given `LICENCES` where each entry is (may change, may sell, must share changes). A use of "change" needs may change; "sell" needs may sell; "use as is" is always allowed. For each (licence, use) in `requests`, print `<licence> / <use>: yes` or `: no`. If the licence is not in `LICENCES`, print `<licence> / <use>: unknown licence`. At the end print `allowed: <n> of <total>`.

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

# licence: (may change, may sell, must share changes)
LICENCES = {"MIT": (True, True, False), "GPL": (True, True, True), "All rights reserved":
(False, False, False)}
requests = [
    ("MIT", "change"),
    ("MIT", "sell"),
    ("GPL", "change"),
    ("All rights reserved", "change"),
    ("All rights reserved", "use as is"),
    ("CC BY", "sell"),
]
```

The hint students can ask for: If the licence is not in the table, say so. Otherwise look up what that licence allows and check it against the use asked for, remembering that using it unchanged is always allowed. Count the ones you allowed.

A solution

```
from bugbot import *
connect()
LICENCES = {"MIT": (True, True, False), "GPL": (True, True, True), "All rights reserved":
(False, False, False)}
requests = [
    ("MIT", "change"),
    ("MIT", "sell"),
    ("GPL", "change"),
    ("All rights reserved", "change"),
    ("All rights reserved", "use as is"),
    ("CC BY", "sell"),
]

def allowed(licence, use):
    change, sell, share = LICENCES[licence]
    if use == "change":
        return change
    if use == "sell":
        return sell
    return True

yes = 0
for licence, use in requests:
    if licence not in LICENCES:
        print(f"{licence} / {use}: unknown licence")
    elif allowed(licence, use):
        yes = yes + 1
        print(f"{licence} / {use}: yes")
```

```
else:  
    print(f"{licence} / {use}: no")  
print(f"allowed: {yes} of {len(requests)}")
```

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