



F6.5 Languages and translators

Robust programs · GCSE · OCR J277 2.5.1, AQA 8525 3.4.4, Edexcel 1CP2
3.3.1 · about 20 min

Name _____

Class _____

Date _____

What this lesson is about

High and low level, machine code and assembly; compilers, interpreters and assemblers.

- High level and low level
- Assembly language
- Translators
- Which does BugBot use?
- An interpreter of your own

Questions

6 marks in all

1. Why must a Python program be translated before it runs? [1 mark]
 - ☐ A A processor only understands its own machine code
 - ☐ B Python is too slow
 - ☐ C To remove comments
 - ☐ D To check spelling
2. Which translator turns assembly language into machine code? [1 mark]
 - ☐ A An assembler
 - ☐ B A compiler
 - ☐ C An interpreter
 - ☐ D An IDE
3. Which translator produces a standalone file that runs without it? [1 mark]
 - ☐ A A compiler
 - ☐ B An interpreter
 - ☐ C An editor
 - ☐ D A debugger
4. Why is an interpreter useful while developing a program? [1 mark]
 - ☐ A It stops at the first error, on the line where it happened
 - ☐ B It makes the finished program run fastest
 - ☐ C It produces an executable file
 - ☐ D It hides the source code

5. Which are features of low-level languages?

[1 mark]

Tick every answer that is true.

- ☐ A One instruction for each processor instruction
- ☐ B Written for a particular kind of processor
- ☐ C Close control of memory and hardware
- ☐ D Easy to read and move between computers

6. The robot assembly program is LOAD 4, FWD 20, TURN 90, BEEP 660, DEC, JNZ 1, HALT. How many beeps does it play?

[1 mark]

The task: robot assembly

Finish the interpreter so it runs the robot assembly program below, which uses `LOAD`, `DEC` and `JNZ` to drive a square with a beep at each corner. Store the value of `R` in a variable, and make `JNZ` change the program counter.

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

program = ["LOAD 4", "FWD 20", "TURN 90", "BEEP 660", "DEC", "JNZ 1", "HALT"]

pc = 0
while True:
    op, *args = program[pc].split()
    if op == "FWD":
        forward(60, distance=int(args[0]))
    elif op == "TURN":
        turn_right(30, angle=int(args[0]))
    elif op == "HALT":
        break
    pc = pc + 1
```

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



Do it on the robot

www.bugbotlab.com/learn/f6-5-languages-and-translators/

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

Challenges

1. Add a `WAIT n` instruction, and write an assembly program that flashes a beep three times.
2. Add `JMP a`, an unconditional jump. What happens if a program jumps back with no way to stop?
3. Write your own compiler of sorts: a function that turns the assembly program into a Python program as text, and print the text.