



A13.2 Function types and function application

Functional programming · A level · AQA 7517 4.12.1.1 · about 20 min

Name _____

Class _____

Date _____

What this lesson is about

Functions as mappings, function type $f: A \rightarrow B$, domain and co-domain, and function application with arguments from a Cartesian product.

- A function is a mapping
- Function type, domain and co-domain
- Writing the type in code
- Function application
- Applying a type to the robot

Questions

6 marks in all

1. A function has type $f: A \rightarrow B$. What is B called? [1 mark]

- ☐ A The co-domain
- ☐ B The domain
- ☐ C The argument type
- ☐ D The Cartesian product

2. `is_even` takes a whole number and returns True or False. What is its function type? [1 mark]

- ☐ A `is_even: integer  $\rightarrow$  Boolean`
- ☐ B `is_even: Boolean  $\rightarrow$  integer`
- ☐ C `is_even: integer  $\times$  Boolean  $\rightarrow$  integer`
- ☐ D `is_even: integer  $\rightarrow$  integer`

3. Which statements about a function $f: A \rightarrow B$ are true? [1 mark]

Tick every answer that is true.

- ☐ A A and B are subsets of objects in some data type
- ☐ B Each value in A is mapped to exactly one value in B
- ☐ C Every value in B must be the result for some value in A
- ☐ D A is the set of results the function gives

4. `add` has type `integer  $\times$  integer  $\rightarrow$  integer`. What is `integer  $\times$  integer` called? [1 mark]

5. In function application, how many arguments does `add(3, 4)` strictly take, if `add: integer × integer → integer`? [1 mark]
- ☐ A One, the pair (3, 4)
 - ☐ B Two, 3 and 4
 - ☐ C None, it is a constant
 - ☐ D Three, 3, 4 and the result
6. What does this program print? [1 mark]

```
def compass(degrees):  
    return "NESW"[((degrees + 45) % 360) // 90]  
  
print(compass(44), compass(45), compass(224), compass(315))
```

The task: the compass type

Write `compass` with its type hints, exactly as `def compass(degrees: int) -> str: .` Its domain is the whole numbers 0 to 359, and it returns "N" for 0 to 44 and 315 to 359, "E" for 45 to 134, "S" for 135 to 224 and "W" for 225 to 314. Then: 1. Apply `compass` to **every** value in its domain, collect the different results, and print them sorted alphabetically and separated by spaces, as `range: <letters> .` Work the letters out; do not type them into the `print` . 2. Turn the robot right by 90 degrees four times. After each turn, print `facing <letter>` , where the letter is `compass` applied to the robot's `heading()` rounded to a whole number and taken MOD 360.

```
# the two lines every program starts with: the commands, then the robot  
from bugbot import *  
connect()  
  
def compass(degrees):  
    return "N"  
  
print("range:", compass(0))
```

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



Do it on the robot

www.bugbotlab.com/learn/a13-2-function-types-and-application/

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

Challenges

1. Give the type, domain and co-domain of a function `quadrant(x, y)` that returns 1, 2, 3 or 4 for a point on the mat.
2. Write `compass8(degrees: int) -> str` with the eight points N, NE, E, SE, S, SW, W, NW. What changes, and what stays the same?
3. `battery()` returns a whole percentage. Write down a sensible function type for it. Why is it not really a function in the mathematical sense?