Individual Exercise – Functions and Higher-Order Functions

Part 1 — A simple function

Write a function that converts temperatures from Celsius to Fahrenheit.
The formula is: F = (9/5) * C + 32

Task:
1. Define a function celsius_to_fahrenheit(celsius) that returns the converted value.
2. Ask the user for a temperature in Celsius.
3. Call the function and print the result in a friendly way.

Example output:

```
Enter temperature in Celsius: 20
20°C = 68.0°F
```

Hints:
- Use input() to get data from the user.
- Remember to convert it to float() before calculation.

Part 2 — Your own higher-order function

Now let's create your first higher-order function — one that takes another function as a parameter.
We'll use the celsius_to_fahrenheit function from Part 1 inside it.

Task:
1. Define a function called apply_and_show(func, value) that:
   - Takes a function func and a numeric value value,
   - Calls the function on that value,
   - Prints the result in the format:
     Applying function <name> to <value> gives <result>
   - Returns the result.
2. Use it with your previous celsius_to_fahrenheit function.

Example output:

```
Applying function celsius_to_fahrenheit to 20.0 gives 68.0
```

Bonus challenge (optional):
Write one more small function, for example:

```
def double(x):
   return x * 2
```

Then test:

```
apply_and_show(double, 10)
```

Expected skills after this exercise

- You understand the structure of a normal function (input → processing → output).
- You can pass a function as a parameter — treating it as a first-class citizen.
- You see how higher-order functions make Python flexible and elegant.