Programmation fonctionnelle 2026-2027

Feuille 1 : Premiers pas

Exercice 1.1 For each of the following expressions, indicate whether it is correct ; if so, give its value and type.

1. `12 + 30`
2. `12.0 + 30`
3. `12.0 + 30.0`
4. `int_of_float 12.5 + 30`
5. `float_of_int 12 +. 30.0`

Exercice 1.2 Give the value of each of the following Boolean expressions :

1. `3 = 4 || 4 = 4`
2. `3 = 4 && 4 = 4`
3. `not (3 = 4) && 4 = 4`
4. `not (3 = 4 || 4 = 4)`

Exercice 1.3 For each of the following expressions, indicate whether it is true ; if so, give its value and type.

1. `12 + 30 = 10 + 32`
2. `12.0 + 30.0 = 10 + 32`
3. `12.0 +. 30.0 = float_of_int (10 + 32)`
4. `12.0 +. 30. = 42. && 3 + 4 = 4 + 4`
5. `12.0 +. 30. = 42. && 3 + 4 != 4 + 4`
6. `12.0 +. 30. = 42. || 3 + 4 = 4 + 4`
7. `12.0 +. 30. = 42. && not (3 + 4 = 4 + 4)`
8. `int_of_string "12" + int_of_string "30"`
9. `int_of_string "12" + int_of_string "30" = 42`

Exercice 1.4 For each of the following expressions, indicate whether it is true ; if so, give its value and type.

1. `let x = 12.0 in x +. 30.0`
2. `let x = 12.0 in if x +. 30.0 = 42.0 then 42 else 0`
3. `let x = 12.0 in if x +. 30.0 = 42.0 then "42" else 0`

Exercice 1.5 What is the final value of `result` ?

```
let result =
  let x = 4 in
  let y =
    let x = 3 in
    x + 2
```

```
  in
  let z =
    let y = 10 in
    x + y
  in
  x + y + z
;;
```

Exercice 1.6 For each of the following two mathematical expressions depending on a floating-point number f, write an `OCaML` expression that computes it using the square root and the logarithm functions exactly once. To do this, factorize the calculations of $\ln f$, $\sqrt{f}$, and $\ln sqrtf$ using `let in` expressions.

$$(\sqrt{f} + \ln f) * (\sqrt{f} - 2\ln f)$$

$$(\sqrt{f} + \ln \sqrt{f}) * (\sqrt{f} - 2\ln \sqrt{f})$$

Exercice 1.7 Provide the expression for an anonymous function that doubles its argument. Provide an example of a call to this function.

Exercice 1.8 Indicate, among the following `OCaML` statements, which ones are :

- an expression ; in this case, give its type and value.
- a 'let' statement (variable-value binding). In this case, give the name, type, and value of the variable.
- a statement that is incorrect due to a syntax error. In this case, explain why the statement is syntactically incorrect.
- a statement that is incorrect due to a type error. In this case, explain the type error.

1. `let y = let x = 12.0 in x +. 30.0`
2. `let y = let x = 12.0 in if x +. 30.0 then 42 else 0`
3. `let x = 3 in let y = 4 in x + y`
4. `let z = let x = 3 in let y = 4 in x + y`
5. `let x = 3 in let y = 4`

Exercice 1.9 Write an expression dependent on a variable `i` that returns a string : `"positive"` if `i` is strictly positive, `"negative"` if `i` is strictly negative, `"null"` otherwise.

Exercice 1.10 Same questions as for the exercise 1.8.

1. `fun x -> x`
2. `(fun x -> x) 5`
3. `fun x -> x + 1`
4. `let f x y = x - y in f (f 1 1) 1`
5. `let compose = fun f g -> fun x -> f (g x)`
6. `let compose f g = fun x -> f (g x)`
7. `let compose f g x = f (g x)`
8. `let compose = fun f g x -> f (g x)`
9. `let mystere =`

```
let square = fun x -> x * x in
let compose = fun f g -> fun x -> f (g x) in
compose square square
```