SIMPLE INTEGER EXPRESSIONS (3/3 points)

What is the result of compiling and evaluating `1 + 2 * 3 - 4`?

- ○ Syntax error.
- ○ Type error.
- ○ `-3`
- ○ `-1`
- ◉ `3` ✔
- ○ `5`

As most programming languages, OCaml follows the **usual priority rules** in arithmetic expressions. This expression is read as: `1 + (2 * 3) - 4`.

What is the result of compiling and evaluating `10 / 3 * 3`?

- ○ Syntax error.
- ○ Type error.
- ○ `1`
- ○ `1.111111`
- ◉ `9` ✔
- ○ `10`

- Exceptions, input/output and imperative constructs
- Modules and data abstraction

- The `/` and `*` operators are **left associative**. It means that this expression is read as: `(10 / 3) * 3`.
- The `/` operator is the **integer division**.

What is the result of compiling and evaluating `42 + 73 mod 5 * 2`?

- Syntax error.
- Type error.
- `0`
- `5`
- `10`
- `90`
- `48` ✔

The `mod` operator computes the **rest of the integer division**. It is one of the rare infix operators composed only of alpha-numerical characters. It has a higher priority than other arithmetic operators, and this expression is read as: `42 + ((73 mod 5) * 2)`.

Vous avez utilisé 1 essais sur 3

SIMPLE INTEGER EXPRESSIONS (BIS) (3/3 points)

Warning: you only have 1 attempt (but anyway the result will not count in the final grading).

What is the result of compiling and evaluating `2 + 3 * 4 - 5`?

- Syntax error.
- Type error.
- `-5`
- `-1`
- `9` ✔
- `15`

This expression is read as: `2 + (3 * 4) - 5`.

What is the result of compiling and evaluating `5 * 4 / 3 - 2`?

- Syntax error.
- Type error.
- `3`
- `4` ✔
- `4.66666`
- `20`

This expression is read as: `((5 * 4) / 3) - 2`, and the integer division of `20` by `3` is `6`.

What is the result of compiling and evaluating `11 / 4 * 4 + 11 mod 4`?

- Type error.
- `3`
- `3.6875`
- `11` ✔
- `14`

This expression is read as: `((11 / 4) * 4) + (11 mod 4)`, which is equal to `(2 * 4) + 3`.

Vous avez utilisé 1 essais sur 1