Overview course Programming 1

1. Variables 1
2. Variables 2
3. Variables 3
4. Conditionals
5. Iterations
6. Functions 1
7. Functions 2
8. Arrays
9. Strings – Game
10. Classes 1 – Encapsulation
11. Classes 2 – Static const

DIGITAL ARTS & ENTERTAINMENT

Conditional Expressions

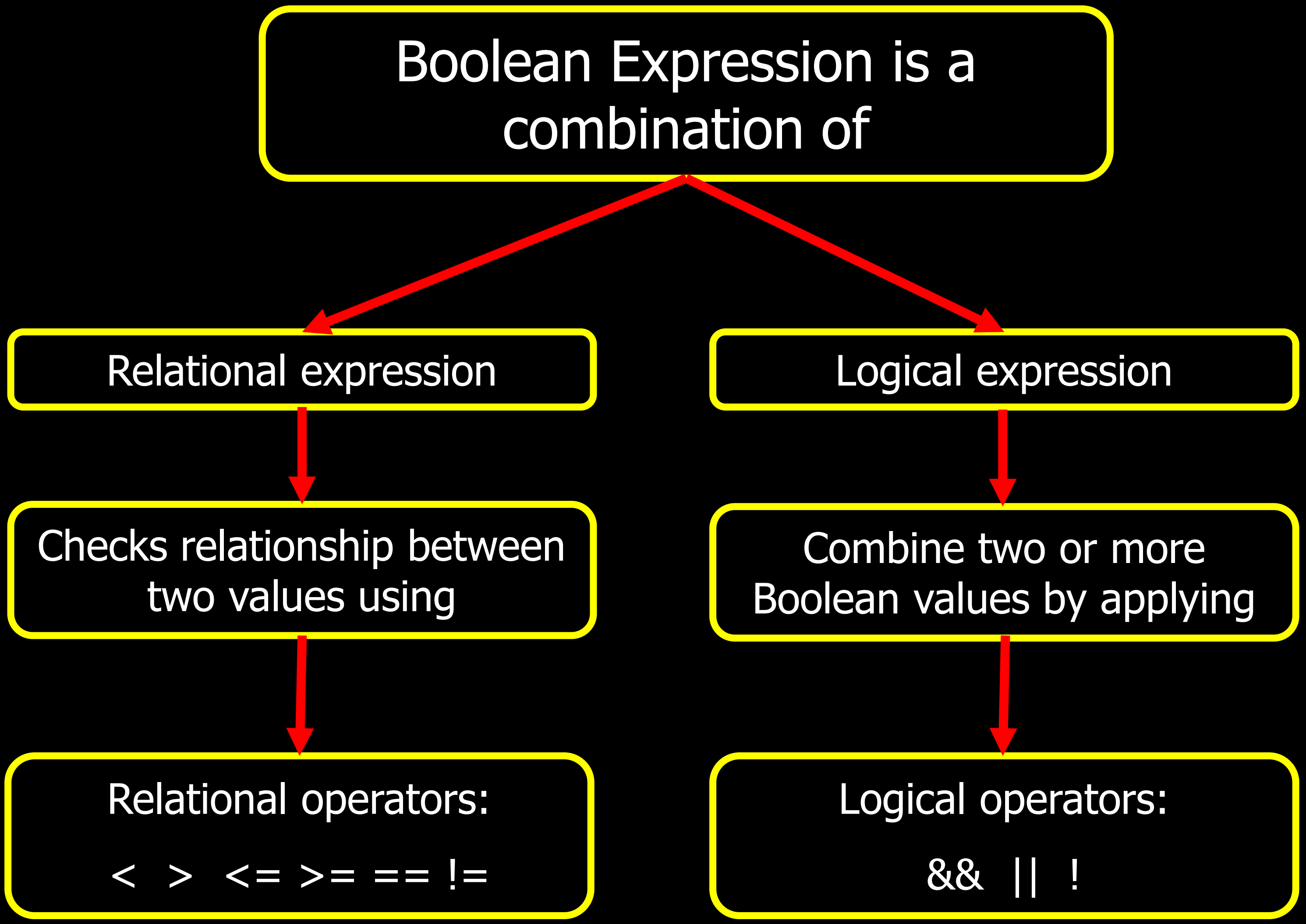
howest
De Hogeschool West-Vlaanderen
Lid van de Associatie Universiteit Gent

The *bool* datatype

- C++: *true* or *false* > not 0 or 1.
- Datatype bool:
  - Primitive type.
  - Can only contain the values *true* and *false*. Eg:

```
bool g_CanIDraw{ true };
```

Boolean expression


Boolean Expression is a combination of
Relational expression
Logical expression
Checks relationship between two values using
Combine two or more Boolean values by applying
Relational operators:
< > <= >= == !=
Logical operators:
&& || !

Boolean expression

- Expression: "anything you can write that results in a value".
  - E.g.   5 + 7
- Boolean expression: an expression where the result is either *true* or *false*.
  - E.g. 5 == 2 + 3

Boolean expression

- Can contain combinations of *relational* operators and *logical* operators. Eg.

```
(a >= b && c < d ) || (b == c)
```

- *relational* operators have precedence on *logical* operators.
  - Precedence: http://en.cppreference.com/w/cpp/language/operator_precedence

Relational expression

- Checks the relationship between two values by applying the *relational operators* :

<	Smaller than
>	Greater than
<=	Smaller than or equal to
>=	Greater than or equal to
==	Is equal to
!=	Is not equal to

Relational expression

- The following conditions hold:

```
int g_Year{ 1850 };
g_Year < 1900    -> ?
g_Year == 1850   ->
g_Year != 1850   ->
g_Year >= 1800   ->
g_Year <= 1850   ->
```

Relational expression

- The following conditions hold:

```
int g_Year{ 1850 };
g_Year < 1900    -> true
g_Year == 1850   -> ?
g_Year != 1850   ->
g_Year >= 1800   ->
g_Year <= 1850   ->
```

Relational expression

- The following conditions hold:

```
int g_Year{ 1850 };
g_Year < 1900    -> true
g_Year == 1850   -> true
g_Year != 1850   -> ?
g_Year >= 1800   ->
g_Year <= 1850   ->
```

Relational expression

- The following conditions hold:

```
int g_Year{ 1850 };
g_Year < 1900    -> true
g_Year == 1850   -> true
g_Year != 1850   -> false
g_Year >= 1800   -> ?
g_Year <= 1850   ->
```

Relational expression

- The following conditions hold:

```
int g_Year{ 1850 };
g_Year < 1900    -> true
g_Year == 1850   -> true
g_Year != 1850   -> false
g_Year >= 1800   -> true
g_Year <= 1850   -> ?
```

Relational expression

- The following conditions hold:

```
int g_Year{ 1850 };
g_Year < 1900    -> true
g_Year == 1850   -> true
g_Year != 1850   -> false
g_Year >= 1800   -> true
g_Year <= 1850   -> true
```

Boolean expression

- Can contain combinations of *relational* operators and *logical* operators. Eg.

```
(a >= b && c < d ) || (b == c)
```

Logical expression

- Combine two or more Boolean values by applying the *logical operators* :

&&	Logical AND
||	Logical OR
!	Logical NOT

Logical expression: AND

- Logical AND: &&
  - true && true → true
  - true && false → false
  - false && true → false
  - false && false → false

Logical expression: OR

- Logical OR: ||
    - true || true → true
    - true || false → true
    - false || true → true
    - false || false → false

Logical expression: NOT

- Logical NOT: !
    - !true → false
    - !false → true

Logical expression

- If g_Year has a value of 1850, then the result of the following expression is:

```
! ( g_Year == 1995 || g_Year == 2001 )
              ↓                 ↓
      ! ( false       ||      false      )
                      ↓
    ! (             false                 )
                      ↓
                     true
```

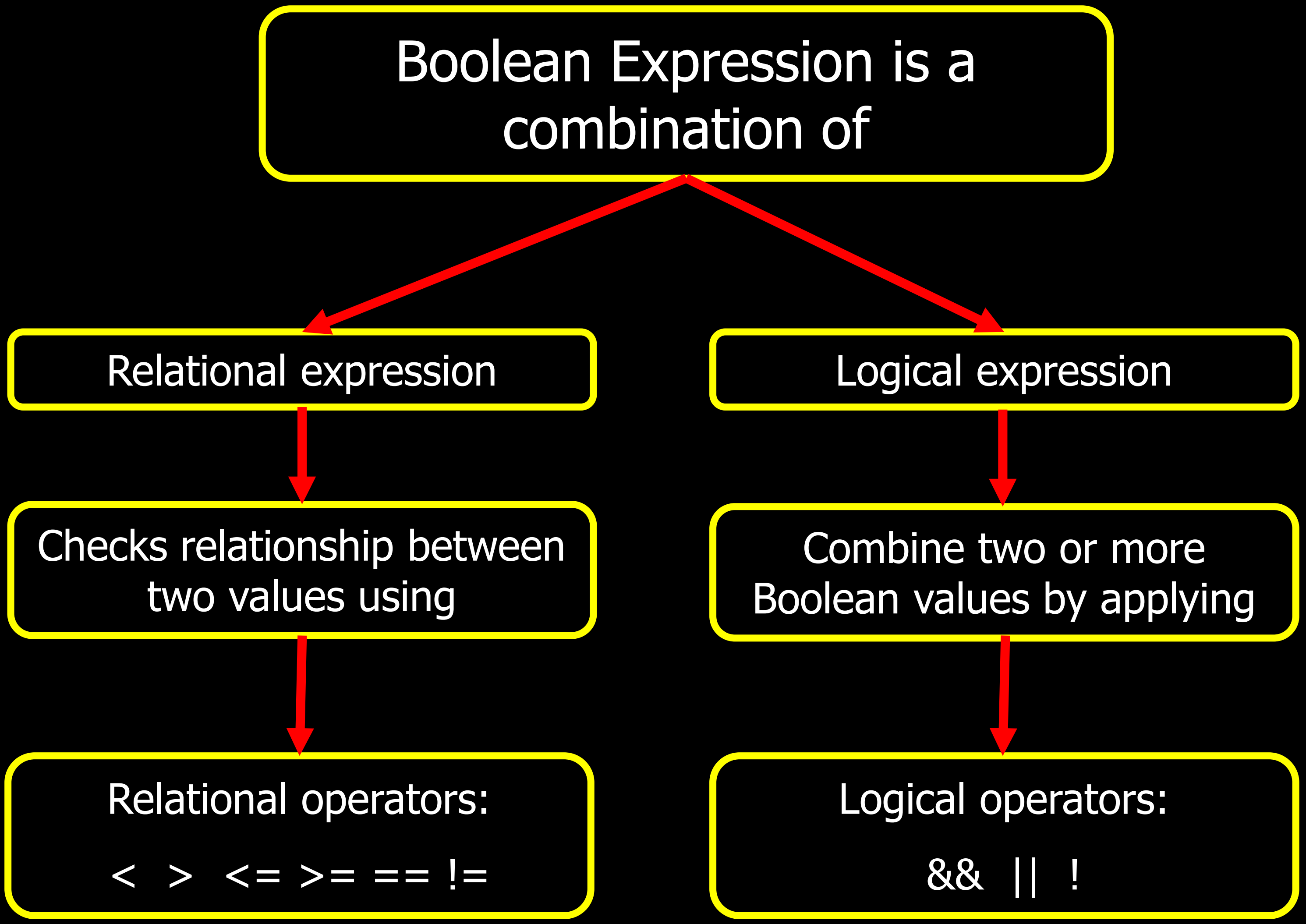
Careful!

- Don’t do:

```
! ( g_Year == 1995 || 2001 )
```

It doesn’t do what you think…
Don’t do it.

Boolean expression


Boolean Expression is a combination of
Relational expression
Logical expression
Checks relationship between two values using
Combine two or more Boolean values by applying
Relational operators:
< > <= >= == !=
Logical operators:
&& || !