Fundamentals of Programming I

3A The Ternary Operator ?

SUPPLEMENT I

Grado en Ingeniería Informática

Luis Hernández Yáñez
Facultad de Informática
Universidad Complutense

The Ternary Operator ?

→ *Condition* → ? → *Exp1* → : → *Exp2* →

Conditional Expression

Two alternatives

- *Condition*: Logical expression
- *Exp1* and *Exp2*: Expressions

If *Condition* evaluates to `true`, the result is *Exp1*
If *Condition* evaluates to `false`, the result is *Exp2*

```
int a = 5, b = 3, c;
c = (a + b == 10) ? 2 : 3;
c = (  8   == 10) ? 2 : 3;
c =     false     ? 2 : 3;
c = 3;
```

Operators (Priority)
++ -- (postfix) Function Call Casts
++ -- (prefix) ! - (sign change)
* / %
+ -
< <= > >=
== !=
&&
||
?:
= += -= *= /= %=

The Ternary Operator ?

Equivalent `if-else`

```
c = (a + b == 10) ? 2 : 3;
```

Is equivalent to:

```
if (a + b == 10)
   c = 2;
else
   c = 3;
```

Can be nested:

```
cout << (grade == 10 ? "MH" : (grade >= 9 ? "SB" :
(grade >= 7 ? "NT" : (grade >= 5 ? "AP" : "SS"))))
```

The equivalent `if-else-if` ladder is in the next page...

The Ternary Operator ?

Equivalent `if ... else if ...` ladder

```
cout << (grade == 10 ? "MH" : (grade >= 9 ? "SB" :
(grade >= 7 ? "NT" : (grade >= 5 ? "AP" : "SS"))))
```

```
if (grade == 10)
   cout << "MH";
else if (grade >= 9)
   cout << "SB";
else if (grade >= 7)
   cout << "NT";
else if (grade >= 5)
   cout << "AP";
else
   cout << "SS";
```