

Short-Circuit Operators

Strictly, && and || are applied in a different way:

a && b : If a is false then b is not evaluated -result is false

a || b : if a is true then b is not evaluated -result is true

Operator Precedence

See Text book Appendix C, page 472 (4th ed.)

!

* / %

+ -

&&

||

Brackets ‘(...)’ have the strongest precedence.

Primitive Types vs. Object Types

Primitive Type

A variable of a primitive type (basic type) stores a value directly in memory:

```
int i;  
i = 32;
```

i 32

Other primitive types are:

double, float, boolean, char, byte, short, long

e.g. `char ch = 'a' ;`
`double num = 54.3`

See Appendix B
P469 (4th ed.)

Method Calls

- External method calls - methods can call methods of other objects using the “dot” notation:

objectName.methodName(parameter-list)

e.g. `minutes.increment()` ;

- The method is sought in class of object referenced by minutes i.e. **NumberDisplay**
- `increment()` must be **public** (or protected) in class **NumberDisplay**

Method Calls

In class ClockDisplay:

```
public void timeTick()  
{  
    minutes.increment();  
    if (minutes.getValue() == 0) {  
        // it just rolled over!  
        hours.increment();  
    }  
    updateDisplay();  
}
```

Which class?

Which class?

Key Points...

- object references
- primitive types
- classes define types e.g. `ClockDisplay`, `NumberDisplay`
- object creation using `new` and constructors
- overloading – more than one constructor
- internal / external method call

Most important overall concept in Chapter 3 is the way that objects cooperate.

**Read the code for `ClockDisplay` and `NumberDisplay`.
We'll be using it in the Practical.**

Next Week Chapter 4: Sections 4.1 – 4.7