Size of the operands used in an assembler instruction

- **Important fact in assembler programming**
  - **Important fact:**

    - ***Every*** **machine (= assembler) instruction** uses a **specific operand size**

  - **In other words**; an **assembler instruction** will ***never*** say:

    - **Add** these 2 values
    - **Subtract** these 2 values

  - An **assembler instruction** will ***always*** say:

    - **Add** these 2 **byte** values **or**
    - **Add** these 2 **short** values **or**
    - **Add** these 2 **int** values **or**
    - And so on

    - **Subtract** these 2 **byte** values **or**
    - **Subtract** these 2 **short** values **or**
    - **Subtract** these 2 **int** values **or**
    - And so on

- **Denoting the operand size in an M68000 assembler instruction**
  - The **M68000** has a ***very*** **simple way** to **denote** the ***size*** of the **operand** in an **instruction/operation**:

```
OpCode.b     means:  byte                operands
OpCode.w     means:  word (2 bytes)      operands
OpCode.l     means:  long word (4 bytes) operands
```

Example:

- The **nmenomic** for the **addition** operation in **m68000** is:

```
    add    operand1, operand2
```

Therefore:

```
  add.b   operand1, operand2    means: add 2 byte operands
  add.w   operand1, operand2    means: add 2 word operands
  add.l   operand1, operand2    means: add 2 long word operands
```

Note:

- To **specify** **where** to **find** an **operand**:
    - Use the ***addressing modes*** that you have **learned** previously !!!!