



東北大學
Northeastern University

软件工程

张爽

东北大学软件学院





7.2

Good Programming Practice

Consistent & Meaningful Variable Names

◆ “*Meaningful*” to future maintenance programmer

➤ **ENGLISH**

- Module contains variables *freqAverage*, *frequencyMaximum*, *minFr*, *frqncyTotl* .
- Maintenance programmer has to know if *freq*, *frequency*, *fr*, *frqncy* all refer to the same thing.

Consistent & Meaningful Variable Names

- ◆ “*Consistent*” to aid maintenance programmer
- Can use *frequencyAverage*, *frequencyMaximum*, *frequencyMinimum*, *frequencyTotal*
- Can also use *averageFrequency*, *maximumFrequency*, *minimumFrequency*, *totalFrequency*

Issue of Self-Documenting Code

- ◆ **Exceedingly rare**
- ◆ **Can module be understood easily and unambiguously by**
 - **The programmer some times later ?**
 - **SQA team ?**
 - **Maintenance programmers ?**
 - **All others who have to read the code ?**

Prologue Comments

◆ Mandatory at top of every single module

- **Module name**
- **Brief description of what the module does**
- **Programmer's name**
- **Date module was coded**
- **Date module was approved, and by whom**
- **Module parameters**
- **Variable names, in alphabetical order, and uses**
- **Files accessed/ updated by this module**
- **Module i/o**
- **Error handling capabilities**
- **Name of file of test data (for regression testing)**
- **List of modifications made, when, approved by whom**
- **Known faults, if any**



Inline Comments

◆ Suggestion

- **Comments are essential whenever code is written in a non-obvious way.**

.....

/ the following statements are to determine
the customer's level */*

.....



Code layout

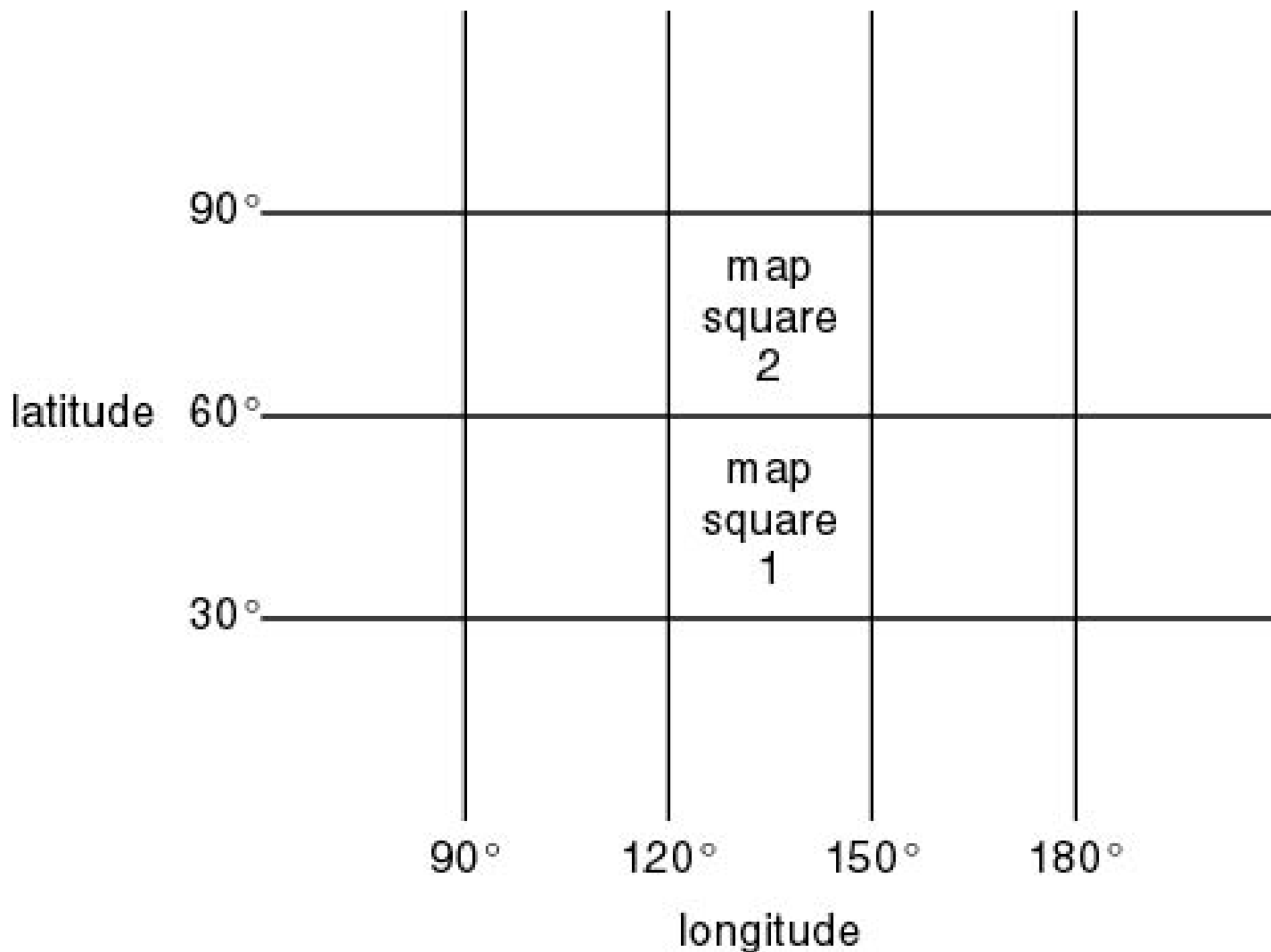
◆ Code layout for increased readability

- Use indentation
- Use blank lines
- Use white space



Nested *if* Statements

◆ Example





Nested *if* Statements

➤ Solution 1. Badly formatted

```
if (latitude > 30 && longitude > 120) {if (latitude <= 60 && longitude <= 150)
mapSquareNo = 1; else if (latitude <= 90 && longitude <= 150) mapSquareNo = 2
else print "Not on the map";} else print "Not on the map";
```



Nested *if* Statements

➤ Solution 2. Well-formatted, badly constructed

```
if (latitude > 30 && longitude > 120)
{
    if (latitude <= 60 && longitude <= 150)
        mapSquareNo = 1;
    else if (latitude <= 90 && longitude <= 150)
        mapSquareNo = 2
    else
        print "Not on the map";
}
else
    print "Not on the map";
```

Nested *if* Statements

➤ Solution 3. Acceptably nested

```
if (longitude > 120 && longitude <= 150 && latitude > 30 && latitude <= 60)
    mapSquareNo = 1;
else if (longitude > 120 && longitude <= 150 && latitude > 60 && latitude <= 90)
    mapSquareNo = 2;
else
    print "Not on the map";
```



Nested *if* Statements

- Combination of *if-if* and *if-else-if* statements is usually difficult to read.

if $\langle \text{condition1} \rangle \ \&\& \ \langle \text{condition2} \rangle$

is better than

if $\langle \text{condition1} \rangle$

if $\langle \text{condition2} \rangle$

- Rule of thumb
 - *if* statements nested to a depth of greater than three should be avoided as poor programming practice.