Code Style

Randy Shepherd
NYU
Correctness is not Sufficient

• In mathematics, its simply important to get the right answer.
• In software engineering, this is not enough.
• Writing software is part science and part craft!
Programming Languages are for Humans

- The reason programming languages were invented was to make working with computers easier for humans.
- Not only when writing, but, more importantly when reading!
- “Always code as if the person who ends up maintaining your code is a violent psychopath who knows where you live.”
  - Jeff Atwood “Coding Horror”
    
    http://www.codinghorror.com/blog/2008/06/coding-for-violent-psychopaths.html
Good Style...

- Makes you a better programmer
- Reduces bugs in your code
- Reduces debugging time
- Reduces cognitive overhead
- Improves your grades
- Earns the love and adoration of your teammates and product owners
What is 'Good Style'?

- Good style is a subjective matter, and is difficult to define.
- The following are usually considered as part of style:
  - layout & indentation
  - thoughtful, consistent naming
  - useful comments
  - sensible modules
Layout & Indentation

Compare:

```java
if (hours < 24 && minutes < 60 && seconds < 60) {
    return true;
} else {
    return false;
}
```

or

```java
if (hours < 24 && minutes < 60 && seconds < 60) {
    return true;
} else {
    return false;
}
```

with something like

```java
if (hours < 24
    && minutes < 60
    && seconds < 60)
{return true
};
else
{return false
};
```
Naming

• Bad names
  → int tmp;
  → getInfo()
  → DataBean

• Good names
  → int clickCounter;
  → calculateAvg()
  → ProductCatalog
Comments

What are good comments?

→ Don't write them. (Your code is self-documenting!)
→ Don't write 'what' code is doing. Write 'why'.
→ Get to the point. Don't write your life story.
→ Alert readers to important code blocks.
Sensible Modules

- What are 'sensible modules'
  → Each language is different

- As an example, Java:
  → If a class file exceeds 200 lines of code your terror alert status should be at ‘elevated’.
  → As it passes 350, it should raise to ‘high’.
  → Past 500 the terror alert is ‘severe’.

- As the number of lines grows, the probability that you are conflating responsibilities of what should be separate modules does too.
In Conclusion

- Code should be easy to read and neat.
- Code should be consistently formatted and organized intelligently.
- Things should be named thoughtfully.
- Code should be *self-documenting*.
- Comments should explain *why* not *what*.