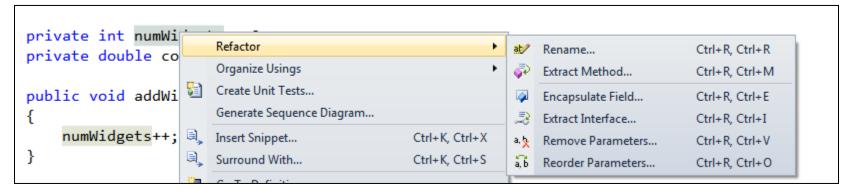
- Think carefully and don't settle
 - Try different names until you're happy
 - Learn to use built-in IDE refactoring features



- Intention-revealing names
 - Why? What? How? Units?

```
int d; // elapsed time in days

int elapsedTime;

int elapsedTimeInDays;
```

Avoid disinformation

```
HashSet<Account> accountList;
long accountName;
String numWidgets;
HashSet<Account> accountGroup;
long accountNumber;
```

```
int a = 1;
if ( 0 == 1 )
    a = 01;
else
    1 = 01;
```

- Make meaningful distinctions
 - Avoid noise words that add anything
 - Reader should be able to determine difference between similar items

```
int count;
int theCount;
int aCount;

double amountVariable;
String nameString;
Customer custObject;

getActiveAccount();
getActiveAccounts();
getActiveAccountInfo();
```

- Use pronounceable names
 - "If you can't pronounce it, you can't discuss it without sounding like an idiot"

```
class DtaRcrd102
{
   private Date genymdhms;
   private Date modymdhms;
   private final String pszqint = "102";
}
```

```
class Customer
{
   private Date generationTimestamp;
   private Date modificationTimestamp;
   private final String recordId= "102";
}
```

Avoid encodings

- e.g. Hungarian notation, member prefixes
- Has become unnecessary
 - Modern IDEs, strongly typed languages, trend towards small classes and methods

```
boolean bBusy;
int iListSize;
PhoneNumber strPhone;

private int m_iLastItem;
m_iLastItem++;
```

```
boolean isBusy;
int listSize;
PhoneNumber phone;

private int lastItem;
lastItem++;
this.lastItem++;
```

Be consistent

- Pick one word per concept
 - Use fetch, retrieve or get, not all of them
- If plural, probably should contain multiple things

```
fetchFullPage()
retrieveHeader()
getFooter()

setTemp(double newTemp)
updateTime(double newTime)

double scores;
int [] score;
```

```
fetchFullPage()
fetchHeader()
fetchFooter()

setTemp(double newTemp)
setTime(double newTime)

double sumScores;
int [] scores;
```

- Avoid gratuitous context
 - Don't prefix every class name in a project
 - Use a namespace instead

```
class MailingAddress
{
    String addrState;
    long addrZip;
}
```

```
class MailingAddress
{
    String state;
    long zip;
}
```

```
...
String addrState = "MN";
...
```