

Names:

1. Are names clear and brief with clarity taking precedence over brevity? Do they avoid abbreviation and ambiguity?
2. Are names consistent?
3. Does the code avoid magic numbers and string literals?

Form:

1. Is the style consistent and readable?
2. Are comments useful or simply alibis for poor coding?
3. Are standard practices used?
4. Are all lines less than 80 characters? Do methods use 5 or fewer parameters? Do all methods have at most 30 lines?

Function:

1. Does each method have a clear, well-defined purpose? Is that purpose conveyed by the name of the method?
2. Is the level of abstraction consistent within methods?
3. Do initialization methods begin with superclass initialization?

Economy:

1. Is it simple?
2. Are non-obvious algorithms adequately explained?
3. Is the code D.R.Y.

Correctness:

1. Are function parameters tested for validity? Is exception handling used?
2. Are loop bounds correct?

Extensibility:

Is it extensible? (Think of one or two ways it may need to be extended and consider how difficult those extensions would be to implement.)