# Rewriting the original program:

## NAMING

Introduction

Importance of naming, earlier finds (Butler, et al. 2009)

|  |  |  |
| --- | --- | --- |
| Name | Description | Example of flawed identifier(s) |
| Capitalization Anomaly | Identifiers should be appropriately capitalized | HTMLEditorKit, pagecounter |
| Consecutive Underscores | Consecutive underscores should not be used in identifier names | foo\_\_bar |
| Dictionary Words | Identifier names should be composed of words found in the dictionary and abbreviations, and acronyms, that are more commonly used than the unabbreviated form. | strlen |
| Excessive Words | Identifier names should be composed of no more than four words or abbreviations | floatToRawIntBits() |
| Enumeration Identifier Declaration Order | Unless there are compelling and obvious reasons otherwise, enumeration constants should be declared in alphabetical order | enum Card {ACE, EIGHT, FIVE, FOUR, JACK, KING ...} |
| External Underscores | Identifiers should not have either leading or trailing underscores. | \_foo\_ |
| Identifier Encoding | Type information should not be encoded in identi- fier names using Hungarian notation or similar | int iCount; |
| Long Identifier Name | Long identifier names should be avoided where possible | getPolicyQualifiersRe-jected |
| Naming Convention Anomaly | Identifiers should not consist of non-standard mixes of upper and lower case characters. | FOO\_bar |
| Number of Words | Identifiers should be composed of between two and four words. | ArrayOutOfBoundsExcep-tion, name |
| Numeric Identifier Name | Identifiers should not be composed entirely of numeric words or numbers | FORTY\_TWO |
| Short Identifier Name | Identifiers should not consist of fewer than eight characters, with the exception of: c, d, e, g, i, in, inOut, j, k, m, n, o, out, t, x, y, z | name |

Butler et al and comments on finds

What was compared?

What problems were solved, through comparison?

Why was this important?

What was the specific hypothesis?

How did you test this hypothesis?

What insights does comparison reveal? Why specific approach useful?

Method

1: Identify algorithms

2: Explain Implementations

Results / Discussion

## MODULATION/ ENCAPSULATION