# English and LaTeX—Writing Tips

Francis Ting

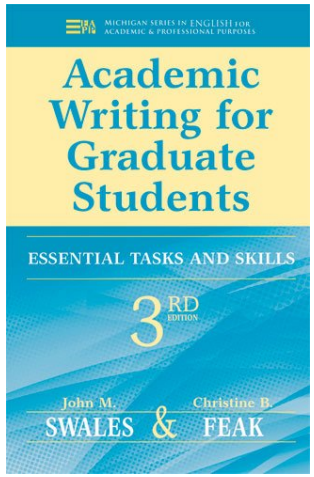
Xi'an Jiaotong-Liverpool University

# Acknowledgement

This slide is is modified by based upon the original slide by Ricardo Terra (rterrabh@gmail.com). His generosity in sharing the LaTeX source files is very much appreciated.

# Main Reference

John Swales and Christine B. Feak. *Academic writing for graduate students: essential tasks and skills*. 3rd ed. University of Michigan Press/ELT, 2012



# Tips

# Tip #1

*He hasn't nothing to do.*

# Tip #1

*He hasn't nothing to do.*

# Tip #1

*He hasn't nothing to do.*
*He has nothing to do. (or)*
*He hasn't anything to do.*

# Tip #1

*He hasn't nothing to do.*
*He has nothing to do. (or)*
*He hasn't anything to do.*

*He isn't neither tall nor handsome.*

# Tip #1

*He hasn't nothing to do.*
*He has nothing to do. (or)*
*He hasn't anything to do.*

*He isn't neither tall nor handsome.*

# Tip #1

*He hasn't nothing to do.*
*He has nothing to do. (or)*
*He hasn't anything to do.*

*He isn't neither tall nor handsome.*
*He is neither tall nor handsome. (or)*
*He isn't either tall or handsome.*

# Tip #1

*He hasn't nothing to do.*
*He has nothing to do. (or)*
*He hasn't anything to do.*

*He isn't neither tall nor handsome.*
*He is neither tall nor handsome. (or)*
*He isn't either tall or handsome.*

*There isn't no way to validate such results.*

# Tip #1

*He hasn't nothing to do.*
*He has nothing to do. (or)*
*He hasn't anything to do.*

*He isn't neither tall nor handsome.*
*He is neither tall nor handsome. (or)*
*He isn't either tall or handsome.*

*There isn't no way to validate such results.*

# Tip #1

*He hasn't nothing to do.*
*He has nothing to do. (or)*
*He hasn't anything to do.*

*He isn't neither tall nor handsome.*
*He is neither tall nor handsome. (or)*
*He isn't either tall or handsome.*

*There isn't no way to validate such results.*
*There is no way to validate such results. (or)*
*There isn't way to validate such results.*

*He hasn't nothing to do.*
*He has nothing to do. (or)*
*He hasn't anything to do.*

*He isn't neither tall nor handsome.*
*He is neither tall nor handsome. (or)*
*He isn't either tall or handsome.*

*There isn't no way to validate such results.*
*There is no way to validate such results. (or)*
*There isn't way to validate such results.*

**Tip #1: Beware Double Negative (4)**

- Double negative is positive!

# Tip #2

*In Table IV, 14 out of 32 recommendations (e.g. D18, D20, D21, A4, etc) include a suggestion to move methods or classes to more suitable modules i.e., better modules.*

# Tip #2

*In Table IV, 14 out of 32 recommendations (e.g. D18, D20, D21, A4, etc) include a suggestion to move methods or classes to more suitable modules i.e., better modules.*

# Tip #2

*In Table IV, 14 out of 32 recommendations (e.g. D18, D20, D21, A4, etc) include a suggestion to move methods or classes to more suitable modules i.e., better modules.*

*In Table IV, 14 out of 32 recommendations (e.g., D18, D20, D21, A4, etc.) include a suggestion to move methods or classes to more suitable modules, i.e., better modules.*

# Tip #2

*In Table IV, 14 out of 32 recommendations (e.g. D18, D20, D21, A4, etc) include a suggestion to move methods or classes to more suitable modules i.e., better modules.*

*In Table IV, 14 out of 32 recommendations (e.g., D18, D20, D21, A4, etc.) include a suggestion to move methods or classes to more suitable modules, i.e., better modules.*

**Tip #2: Using connectors properly (etc1, 16, 11)**

- Using correctly: "e.g.", "i.e.", and "etc."

# Tip #3

*Class A depends on classes $C_1$, $C_2$ and $C_n$.*

# Tip #3

*Class A depends on classes $C_1$, $C_2$ and $C_n$.*

# Tip #3

*Class A depends on classes* $C_1$, $C_2$ *and* $C_n$.
*Class A depends on classes* $C_1$, $C_2$, *and* $C_n$.

# Tip #3

*Class A depends on classes $C_1$, $C_2$ and $C_n$.*
*Class A depends on classes $C_1$, $C_2$, and $C_n$.*

*Tomorrow, I will play guitar, go shopping or die.*

# Tip #3

*Class A depends on classes $C_1$, $C_2$ and $C_n$.*
*Class A depends on classes $C_1$, $C_2$, and $C_n$.*

*Tomorrow, I will play guitar, go shopping or die.*

# Tip #3

*Class A depends on classes $C_1$, $C_2$ and $C_n$.*
*Class A depends on classes $C_1$, $C_2$, and $C_n$.*

*Tomorrow, I will play guitar, go shopping or die.*
*Tomorrow, I will play guitar, go shopping, or die.*

# Tip #3

*Class A depends on classes $C_1$, $C_2$ and $C_n$.*
*Class A depends on classes $C_1$, $C_2$, and $C_n$.*

*Tomorrow, I will play guitar, go shopping or die.*
*Tomorrow, I will play guitar, go shopping, or die.*

*She likes cooking, and jogging.*

# Tip #3

*Class A depends on classes $C_1$, $C_2$ and $C_n$.*
*Class A depends on classes $C_1$, $C_2$, and $C_n$.*

*Tomorrow, I will play guitar, go shopping or die.*
*Tomorrow, I will play guitar, go shopping, or die.*

*She likes cooking, and jogging.*

# Tip #3

*Class A depends on classes $C_1$, $C_2$ and $C_n$.*
*Class A depends on classes $C_1$, $C_2$, and $C_n$.*

*Tomorrow, I will play guitar, go shopping or die.*
*Tomorrow, I will play guitar, go shopping, or die.*

*She likes cooking, and jogging.*
*She likes cooking and jogging.*

# Tip #3

*Class A depends on classes $C_1$, $C_2$ and $C_n$.*
*Class A depends on classes $C_1$, $C_2$, and $C_n$.*

*Tomorrow, I will play guitar, go shopping or die.*
*Tomorrow, I will play guitar, go shopping, or die.*

*She likes cooking, and jogging.*
*She likes cooking and jogging.*

**Tip #3: Comma in series (16, 14)**

- Using comma between all the items in a series, including the last two
- This comma is known by *serial comma*

# Tip #4

*We come up with such syntactic sugar to make the development task easier.*

# Tip #4

*We come up with such syntactic sugar to make the development task easier.*

# Tip #4

*We come up with such syntactic sugar to make the development task easier.*

*We designed such syntactic sugar to facilitate the development task.*

# Tip #4

*We come up with such syntactic sugar to make the development task easier.*

*We designed such syntactic sugar to facilitate the development task.*

**Tip #4: Avoid phrasal verbs (10)**

- Why? Because it is informal!
- See more phrasal verbs in the next slide

# Tip #5

*Our recommendations were validated by the senior architect responsible for the architecture.*

# Tip #5

*Our recommendations were validated by the senior architect responsible for the architecture.*

# Tip #5

*Our recommendations were validated by the senior architect responsible for the architecture.*

*The senior architect responsible for the architecture validated our recommendations.*

# Tip #5

*Our recommendations were validated by the senior architect responsible for the architecture.*

*The senior architect responsible for the architecture validated our recommendations.*

**Tip #5: Avoid passive voice (19, 12)**

- Passive voice is not usual in English writing as it is in Portuguese
- Can I use sometimes? *Yes, you can*
- When? *The passive voice makes sense because the agent is relatively unimportant compared to the action itself and what is acted upon*
  - "Authorities *make* rules to be broken" $\Longrightarrow$ "Rules *are made* to be broken"
  - "Surgeons successfully *performed* a new experimental liver-transplant operation yesterday" $\Longrightarrow$ "A new experimental liver-transplant operation *was performed* successfully yesterday"

# Tip #6

*He is tall, but poor.*

# Tip #6

*He is tall, but poor.*

# Tip #6

*He is tall, but poor.*
*He is tall but poor.*

# Tip #6

*He is tall, but poor.*
*He is tall but poor.*

*He is tall but his mom is short.*

# Tip #6

*He is tall, but poor.*
*He is tall but poor.*

*He is tall but his mom is short.*

# Tip #6

*He is tall, but poor.*
*He is tall but poor.*

*He is tall but his mom is short.*
*He is tall, but his mom is short.*

*He is tall, but poor.*
*He is tall but poor.*

*He is tall but his mom is short.*
*He is tall, but his mom is short.*

**Tip #6: Correct use of but (16)**

- Use comma before *but* in order to connect two independent clauses
- Otherwise, do not use

*For the first system – a 21 KLOC open-source strategic management system – we could indicate the correct refactoring recommendations for 75% of the detected violations.*

*For the first system – a 21 KLOC open-source strategic management system – we could indicate the correct refactoring recommendations for 75% of the detected violations.*

# Tip #7

*For the first system – a 21 KLOC open-source strategic management system – we could indicate the correct refactoring recommendations for 75% of the detected violations.*

*For the first system—a 21 KLOC open-source strategic management system—we could indicate the correct refactoring recommendations for 75% of the detected violations.*

*For the first system – a 21 KLOC open-source strategic management system – we could indicate the correct refactoring recommendations for 75% of the detected violations.*

*For the first system—a 21 KLOC open-source strategic management system—we could indicate the correct refactoring recommendations for 75% of the detected violations.*

**Tip #7: Using dash properly (16)**

- In English: *blablabla—something—blablabla*
- In Portuguese: *blablabla – algo – blablabla*

# Tip #8

*Furthermore, a detailed description of all architectural refactoring recommendations is available in a companion website*[1].

[1] *http://www.dcc.ufmg.br/ terra/spe2013*

# Tip #8

*Furthermore, a detailed description of all architectural refactoring recommendations is available in a companion website*[1] *.*

[1] *http://www.dcc.ufmg.br/ terra/spe2013*

# Tip #8

*Furthermore, a detailed description of all architectural refactoring recommendations is available in a companion website*[1] *.*

[1] *http://www.dcc.ufmg.br/ terra/spe2013*

*Furthermore, a detailed description of all architectural refactoring recommendations is available in a companion website.*[1]

[1] *http://www.dcc.ufmg.br/ terra/spe2013*

# Tip #8

*Furthermore, a detailed description of all architectural refactoring recommendations is available in a companion website*[1].

[1] `http://www.dcc.ufmg.br/ terra/spe2013`

*Furthermore, a detailed description of all architectural refactoring recommendations is available in a companion website.*[1]

[1] `http://www.dcc.ufmg.br/ terra/spe2013`

**Tip #8: Footnote markers (16)**

- Mostly, after the whole sentence
- Sometimes in the middle of the sentence to avoid misunderstandings:
  - "The Lattix Dependency Manager (LDM) tool[2] provides a simple language to declare design rules that the target system implementation must follow (e.g., **`A cannot—use B`**) and visually represents the detected violations in a DSM."

# Tip #9

*It's common to identify patterns they are following.*

# Tip #9

*It's common to identify patterns they are following.*

# Tip #9

*It's common to identify patterns they are following.*

*It's common to identify patterns they're following.*
*It is common to identify patterns they are following.*

# Tip #9

*It's common to identify patterns they are following.*

*It's common to identify patterns they're following.*
*It is common to identify patterns they are following.*

**Tip #9: Contraction forms (16)**

- Should I use or not? *It depends on the journal or conference you're submitting to*
- However, **never** mixed them up

# Tip #10

*I can not play soccer because I broke my leg.*

# Tip #10

*I can not play soccer because I broke my leg.*

# Tip #10

*I can not play soccer because I broke my leg.*

*I cannot play soccer because I broke my leg.*

*I can not play soccer because I broke my leg.*

*I cannot play soccer because I broke my leg.*

**Tip #10: Cannot or Can not? (7)**

- *cannot* is the ordinary modern way of writing *can not*
- Both *cannot* and *can not* are acceptable spellings
  - but the first is much more usual
- In fact, you would use *can not* when the *not* forms part of another construction such as *not only*:
  - "He can not only sing but also dance."

*The approach cannot suggest recommendations.*

# Tip #11

*The approach cannot suggest recommendations.*

# Tip #11

*The approach cannot suggest recommendations.*

*The approach may not suggest recommendations.* *(possibility)*

# Tip #11

*The approach cannot suggest recommendations.*

*The approach may not suggest recommendations.* *(possibility)*

**Tip #11: Can or May? (1)**

- Use *may* for possibility
- Use *can* for ability

# Tip #12

*The formula must satisfy these criterion.*

# Tip #12

*The formula must satisfy these criterion.*

# Tip #12

*The formula must satisfy these criterion.*

*The formula must satisfy these criteria.*

# Tip #12

*The formula must satisfy these criterion.*

*The formula must satisfy these criteria.*

**Tip #12: Greek nouns? (15, 6)**

- Ends with *on* in case of singular (e.g., *phenomenon*)
- Ends with *a* in case of plural (e.g., *phenomena*)

# Tip #13

*We report two case studies with ArchLint. In a first study, we applied the solution in an industrial-strength information system.*

# Tip #13

*We report two case studies with ArchLint. In a first study, we applied the solution in an industrial-strength information system.*

# Tip #13

*We report two case studies with ArchLint. In a first study, we applied the solution in an industrial-strength information system.*

*We reported two case studies with ArchLint. In a first study, we applied the solution in an industrial-strength information system.*

# Tip #13

*We report two case studies with ArchLint. In a first study, we applied the solution in an industrial-strength information system.*

*We reported two case studies with ArchLint. In a first study, we applied the solution in an industrial-strength information system.*

## Tip #13: Parallelism (16)

- Keep the tense
- More examples:
  - *She likes cooking and to read novels.* $\Longrightarrow$
    *She likes cooking and reading novels.* (or)
    *She likes to cook and read novels.*
- More important, sections *Related Work* and *Background* can be written in either *Simple Past* or *Present Perfect*, but **never** mix tenses up

# Tip #14

*The tool relies on search algorithms, like hill climbing and simulated annealing, to suggest six inheritance-related refactorings.*

# Tip #14

*The tool relies on search algorithms, like hill climbing and simulated annealing, to suggest six inheritance-related refactorings.*

# Tip #14

*The tool relies on search algorithms, like hill climbing and simulated annealing, to suggest six inheritance-related refactorings.*

*The tool relies on search algorithms, such as hill climbing and simulated annealing, to suggest six inheritance-related refactorings.*

# Tip #14

*The tool relies on search algorithms, like hill climbing and simulated annealing, to suggest six inheritance-related refactorings.*

*The tool relies on search algorithms, such as hill climbing and simulated annealing, to suggest six inheritance-related refactorings.*

**Tip #14: Like or such as? (2, 13)**

- *Like* says that what follows is intended as a frame of reference to indicate the group of things you're talking about but is itself not included in the group
- *Such as* means that what follows are examples of the things that are part of the group you're talking about
- More examples:
  - Chuck enjoys desserts such as brownies, cheesecakes, and macaroons
  - Chuck enjoys desserts like brownies, cheesecakes, and macaroons

# Tip #15

*Diamonds that are expensive often elicit forgiveness.*

# Tip #15

*Diamonds that are expensive often elicit forgiveness.*

# Tip #15

*Diamonds that are expensive often elicit forgiveness.*

*Diamonds, which are expensive, often elicit forgiveness.*

# Tip #15

*Diamonds that are expensive often elicit forgiveness.*

*Diamonds, which are expensive, often elicit forgiveness.*

**Tip #15: That or which (16, 5, 21)**

- *That* for a restrictive clause, i.e., it specifically restricts some other part of the sentence
- *Which* for a nonrestrictive clause, i.e., it can be left off without changing the meaning of the sentence
- More examples:
  - Our house that has a red door and green shutters needs painting.
  - Our house, which has a red door and green shutters, needs painting.

# Tip #16

*He doesn't need a car, because he lives downtown.*

# Tip #16

*He doesn't need a car, because he lives downtown.*

# Tip #16

*He doesn't need a car, because he lives downtown.*
*He doesn't need a car because he lives downtown.*

# Tip #16

*He doesn't need a car, because he lives downtown.*
*He doesn't need a car because he lives downtown.*

*Since the museum was closed she went to the shopping.*

# Tip #16

*He doesn't need a car, because he lives downtown.*
*He doesn't need a car because he lives downtown.*

*Since the museum was closed she went to the shopping.*

# Tip #16

*He doesn't need a car, because he lives downtown.*
*He doesn't need a car because he lives downtown.*

*Since the museum was closed she went to the shopping.*
*Since the museum was closed, she went to the shopping.*

*He doesn't need a car, because he lives downtown.*
*He doesn't need a car because he lives downtown.*

*Since the museum was closed she went to the shopping.*
*Since the museum was closed, she went to the shopping.*

**Tip #16: Because or since? (20, 3, 18)**

- Conjunctions—such as because, as, since, while, and though—the dependent clause (the part beginning with the conjunction) can also come at the beginning of the sentence
- In this case (and ONLY in this case), a comma can be used to join the two clauses

# Tip #17

*Although lasagna and pizza are both popular dishes in the school cafeteria, most students prefer the later.*

# Tip #17

*Although lasagna and pizza are both popular dishes in the school cafeteria, most students prefer the later.*

# Tip #17

*Although lasagna and pizza are both popular dishes in the school cafeteria, most students prefer the later.*

*Although lasagna and pizza are both popular dishes in the school cafeteria, most students prefer the latter.*

# Tip #17

*Although lasagna and pizza are both popular dishes in the school cafeteria, most students prefer the later.*

*Although lasagna and pizza are both popular dishes in the school cafeteria, most students prefer the latter.*

*Jill traveled to Aruba, Australia, and Paris, but her favorite vacation was the latter.*

# Tip #17

*Although lasagna and pizza are both popular dishes in the school cafeteria, most students prefer the later.*
*Although lasagna and pizza are both popular dishes in the school cafeteria, most students prefer the latter.*

*Jill traveled to Aruba, Australia, and Paris, but her favorite vacation was the latter.*

# Tip #17

*Although lasagna and pizza are both popular dishes in the school cafeteria, most students prefer the later.*
*Although lasagna and pizza are both popular dishes in the school cafeteria, most students prefer the latter.*

*Jill traveled to Aruba, Australia, and Paris, but her favorite vacation was the latter.*
*Jill traveled to Aruba, Australia, and Paris, but her favorite vacation was the last.*

*Although lasagna and pizza are both popular dishes in the school cafeteria, most students prefer the later.*
*Although lasagna and pizza are both popular dishes in the school cafeteria, most students prefer the latter.*

*Jill traveled to Aruba, Australia, and Paris, but her favorite vacation was the latter.*
*Jill traveled to Aruba, Australia, and Paris, but her favorite vacation was the last.*

### Tip #17: Using former/latter (8)

- *Former* and *latter* should only be used when comparing two items
  - former = first
  - latter = last
- More examples:
  - John likes both dogs and cats, but he really prefers the former over the latter.

*It was design a tool that implement our approach.*

# Tip #18

*It was design a tool that implement our approach.*

# Tip #18

*It was design a tool that implement our approach.*
*It was designed a tool that implements our approach.*

# Tip #18

*It was design a tool that implement our approach.*
*It was designed a tool that implements our approach.*

*Section 1 describe our approach.*

# Tip #18

*It was design a tool that implement our approach.*
*It was designed a tool that implements our approach.*

*Section 1 describe our approach.*

# Tip #18

*It was design a tool that implement our approach.*
*It was designed a tool that implements our approach.*

*Section 1 describe our approach.*
*Section 1 describes our approach.*

# Tip #18

*It was design a tool that implement our approach.*
*It was designed a tool that implements our approach.*

*Section 1 describe our approach.*
*Section 1 describes our approach.*

*It is showed in Figure 1.*

# Tip #18

*It was design a tool that implement our approach.*
*It was designed a tool that implements our approach.*

*Section 1 describe our approach.*
*Section 1 describes our approach.*

*It is showed in Figure 1.*

# Tip #18

*It was design a tool that implement our approach.*
*It was designed a tool that implements our approach.*

*Section 1 describe our approach.*
*Section 1 describes our approach.*

*It is showed in Figure 1.*
*It is shown in Figure 1.*

# Tip #18

*It was design a tool that implement our approach.*
*It was designed a tool that implements our approach.*

*Section 1 describe our approach.*
*Section 1 describes our approach.*

*It is showed in Figure 1.*
*It is shown in Figure 1.*

*We could got more information.*

# Tip #18

*It was design a tool that implement our approach.*
*It was designed a tool that implements our approach.*

*Section 1 describe our approach.*
*Section 1 describes our approach.*

*It is showed in Figure 1.*
*It is shown in Figure 1.*

*We could got more information.*

# Tip #18

*It was design a tool that implement our approach.*
*It was designed a tool that implements our approach.*

*Section 1 describe our approach.*
*Section 1 describes our approach.*

*It is showed in Figure 1.*
*It is shown in Figure 1.*

*We could got more information.*
*We could get more information.*

# Tip #18

*It was design a tool that implement our approach.*
*It was designed a tool that implements our approach.*

*Section 1 describe our approach.*
*Section 1 describes our approach.*

*It is showed in Figure 1.*
*It is shown in Figure 1.*

*We could got more information.*
*We could get more information.*

**Tip #18: Conjugation (9, 16)**

- Pay attention to third-person singular, past participle, etc.

# Tip #19

*The heuristic #1 has found several violations.*

# Tip #19

*The heuristic #1 has found several violations.*

# Tip #19

*The heuristic #1 has found several violations.*
*Heuristic #1 has found several violations.*

# Tip #19

*The heuristic #1 has found several violations.*
*Heuristic #1 has found several violations.*

*The results are illustrated in the Table 3.*

# Tip #19

*The heuristic #1 has found several violations.*
*Heuristic #1 has found several violations.*

*The results are illustrated in the Table 3.*

# Tip #19

*The heuristic #1 has found several violations.*
*Heuristic #1 has found several violations.*

*The results are illustrated in the Table 3.*
*The results are illustrated in Table 3.*

# Tip #19

*The heuristic #1 has found several violations.*
*Heuristic #1 has found several violations.*

*The results are illustrated in the Table 3.*
*The results are illustrated in Table 3.*

*We rely on the function $f$ to find duplicates.*

# Tip #19

*The heuristic #1 has found several violations.*
*Heuristic #1 has found several violations.*

*The results are illustrated in the Table 3.*
*The results are illustrated in Table 3.*

*We rely on the function $f$ to find duplicates.*

# Tip #19

*The heuristic #1 has found several violations.*
*Heuristic #1 has found several violations.*

*The results are illustrated in the Table 3.*
*The results are illustrated in Table 3.*

*We rely on the function f to find duplicates.*
*We rely on function f to find duplicates. (or)*
*We rely on the f function to find duplicates.*

# Tip #19

*The heuristic #1 has found several violations.*
*Heuristic #1 has found several violations.*

*The results are illustrated in the Table 3.*
*The results are illustrated in Table 3.*

*We rely on the function f to find duplicates.*
*We rely on function f to find duplicates. (or)*
*We rely on the f function to find duplicates.*

*The class* **`ArrayList`** *represents a collection.*

# Tip #19

*The heuristic #1 has found several violations.*
*Heuristic #1 has found several violations.*

*The results are illustrated in the Table 3.*
*The results are illustrated in Table 3.*

*We rely on the function f to find duplicates.*
*We rely on function f to find duplicates. (or)*
*We rely on the f function to find duplicates.*

*The class* `ArrayList` *represents a collection.*

# Tip #19

*The heuristic #1 has found several violations.*
*Heuristic #1 has found several violations.*

*The results are illustrated in the Table 3.*
*The results are illustrated in Table 3.*

*We rely on the function f to find duplicates.*
*We rely on function f to find duplicates. (or)*
*We rely on the f function to find duplicates.*

*The class* `ArrayList` *represents a collection.*
*The* `ArrayList` *class represents a collection. (or)*
*Class* `ArrayList` *represents a collection.*

*The heuristic #1 has found several violations.*
*Heuristic #1 has found several violations.*

*The results are illustrated in the Table 3.*
*The results are illustrated in Table 3.*

*We rely on the function f to find duplicates.*
*We rely on function f to find duplicates. (or)*
*We rely on the f function to find duplicates.*

*The class* `ArrayList` *represents a collection.*
*The* `ArrayList` *class represents a collection. (or)*
*Class* `ArrayList` *represents a collection.*

**Tip #19: Using the (16)**

- You don't use *the* when the target object has an identifier

# Tip #20

*Some people like coffee, other prefer tea.*

# Tip #20

*Some people like coffee, other prefer tea.*

# Tip #20

*Some people like coffee, other prefer tea.*
*Some people like coffee, others prefer tea.*

# Tip #20

*Some people like coffee, other prefer tea.*
*Some people like coffee, others prefer tea.*

*Besides this heuristic, we propose others heuristics.*

# Tip #20

*Some people like coffee, other prefer tea.*
*Some people like coffee, others prefer tea.*

*Besides this heuristic, we propose others heuristics.*

# Tip #20

*Some people like coffee, other prefer tea.*
*Some people like coffee, others prefer tea.*

*Besides this heuristic, we propose others heuristics.*
*Besides this heuristic, we propose other heuristics.*

*Some people like coffee, other prefer tea.*
*Some people like coffee, others prefer tea.*

*Besides this heuristic, we propose others heuristics.*
*Besides this heuristic, we propose other heuristics.*

**Tip #20: Other vs. Others (17)**

- In short:
  - *others* = *other* <*noun*>
  - *other* for singular or plural

# Tip #21

*Our approach achieved a precision about 70%.*

# Tip #21

*Our approach achieved a precision about 70%.*

# Tip #21

*Our approach achieved a precision nearby 70%.*

# Tip #21

*Our approach achieved a precision nearby 70%.*

*We evaluate using a lot of systems.*

# Tip #21

*Our approach achieved a precision nearby 70%.*

*We evaluate using a lot of systems.*

# Tip #21

*Our approach achieved a precision nearby 70%.*

*We evaluate using a considerable amount of systems.*

# Tip #21

*Our approach achieved a precision nearby 70%.*

*We evaluate using a considerable amount of systems.*

*The function gets an instance from the factory.*

# Tip #21

*Our approach achieved a precision nearby 70%.*

*We evaluate using a considerable amount of systems.*

*The function gets an instance from the factory.*

# Tip #21

*Our approach achieved a precision nearby 70%.*

*We evaluate using a considerable amount of systems.*

*The function obtains an instance from the factory.*

# Tip #21

*Our approach achieved a precision nearby 70%.*

*We evaluate using a considerable amount of systems.*

*The function obtains an instance from the factory.*

*We argue that four systems are enough.*

# Tip #21

*Our approach achieved a precision nearby 70%.*

*We evaluate using a considerable amount of systems.*

*The function obtains an instance from the factory.*

*We argue that four systems are enough.*

# Tip #21

*Our approach achieved a precision nearby 70%.*

*We evaluate using a considerable amount of systems.*

*The function obtains an instance from the factory.*

*We argue that four systems are sufficient.*

# Tip #21

*Our approach achieved a precision nearby 70%.*

*We evaluate using a considerable amount of systems.*

*The function obtains an instance from the factory.*

*We argue that four systems are sufficient.*

*Violations happen when constraints are not respected.*

# Tip #21

*Our approach achieved a precision nearby 70%.*

*We evaluate using a considerable amount of systems.*

*The function obtains an instance from the factory.*

*We argue that four systems are sufficient.*

*Violations happen when constraints are not respected.*

# Tip #21

*Our approach achieved a precision nearby 70%.*

*We evaluate using a considerable amount of systems.*

*The function obtains an instance from the factory.*

*We argue that four systems are sufficient.*

*Violations occur when constraints are not respected.*

# Tip #21

*Our approach achieved a precision nearby 70%.*

*We evaluate using a considerable amount of systems.*

*The function obtains an instance from the factory.*

*We argue that four systems are sufficient.*

*Violations occur when constraints are not respected.*

*Because of that, violations are not fixed as the system evolves.*

# Tip #21

*Our approach achieved a precision nearby 70%.*

*We evaluate using a considerable amount of systems.*

*The function obtains an instance from the factory.*

*We argue that four systems are sufficient.*

*Violations occur when constraints are not respected.*

*Because of that, violations are not fixed as the system evolves.*

# Tip #21

*Our approach achieved a precision nearby 70%.*

*We evaluate using a considerable amount of systems.*

*The function obtains an instance from the factory.*

*We argue that four systems are sufficient.*

*Violations occur when constraints are not respected.*

*Therefore/In view of such circumstances/Thereupon, violations are not fixed as the system evolves.*

# Tip #21

*Our approach achieved a precision nearby 70%.*

*We evaluate using a considerable amount of systems.*

*The function obtains an instance from the factory.*

*We argue that four systems are sufficient.*

*Violations occur when constraints are not respected.*

*Therefore/In view of such circumstances/Thereupon,*
*violations are not fixed as the system evolves.*

*It can be freely downloaded at:*

# Tip #21

*Our approach achieved a precision nearby 70%.*

*We evaluate using a considerable amount of systems.*

*The function obtains an instance from the factory.*

*We argue that four systems are sufficient.*

*Violations occur when constraints are not respected.*

*Therefore/In view of such circumstances/Thereupon,*
*violations are not fixed as the system evolves.*

*It can be freely downloaded at:*

# Tip #21

*Our approach achieved a precision nearby 70%.*

*We evaluate using a considerable amount of systems.*

*The function obtains an instance from the factory.*

*We argue that four systems are sufficient.*

*Violations occur when constraints are not respected.*

*Therefore/In view of such circumstances/Thereupon,*
*violations are not fixed as the system evolves.*

*It is publicly available at:*

# Tip #21

*Our approach achieved a precision nearby 70%.*

*We evaluate using a considerable amount of systems.*

*The function obtains an instance from the factory.*

*We argue that four systems are sufficient.*

*Violations occur when constraints are not respected.*

*Therefore/In view of such circumstances/Thereupon, violations are not fixed as the system evolves.*

*It is publicly available at:*

*Our approach triggered correct recommendations for 655 of 828 detected violations.*

# Tip #21

*Our approach achieved a precision nearby 70%.*

*We evaluate using a considerable amount of systems.*

*The function obtains an instance from the factory.*

*We argue that four systems are sufficient.*

*Violations occur when constraints are not respected.*

*Therefore/In view of such circumstances/Thereupon, violations are not fixed as the system evolves.*

*It is publicly available at:*

*Our approach triggered correct recommendations for 655 of 828 detected violations.*

# Tip #21

*Our approach achieved a precision nearby 70%.*

*We evaluate using a considerable amount of systems.*

*The function obtains an instance from the factory.*

*We argue that four systems are sufficient.*

*Violations occur when constraints are not respected.*

*Therefore/In view of such circumstances/Thereupon, violations are not fixed as the system evolves.*

*It is publicly available at:*

*Our approach triggered correct recommendations for 655 out of 828 detected violations (79%).*

# Tip #21

*Our approach achieved a precision nearby 70%.*

*We evaluate using a considerable amount of systems.*

*The function obtains an instance from the factory.*

*We argue that four systems are sufficient.*

*Violations occur when constraints are not respected.*

*Therefore/In view of such circumstances/Thereupon, violations are not fixed as the system evolves.*

*It is publicly available at:*

*Our approach triggered correct recommendations for 655 out of 828 detected violations (79%).*

**Tip #21: Better ways to write the same thing (16)**

- Always prefer the more formal way

# Tip #22

*Our approach focuses in long methods.*

# Tip #22

*Our approach focuses in long methods.*

# Tip #22

*Our approach focuses in long methods.*

*Our approach focuses on long methods.*

# Tip #22

*Our approach focuses in long methods.*

*Our approach focuses on long methods.*

**Tip #22: Very used sentences (16)**

- rely on
- based on
- focus on
- with respect to (w.r.t.)

*The function is formalized as follow:*

# Tip #23

*The function is formalized as follow:*

# Tip #23

*The function is formalized as follows:*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has an unique property that differs it from others.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has an unique property that differs it from others.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

*This file is processed offline.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

*This file is processed offline.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

*This file is processed off-line.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

*This file is processed off-line.*

*It allows users to pay tickets on-line.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

*This file is processed off-line.*

*It allows users to pay tickets on-line.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

*This file is processed off-line.*

*It allows users to pay tickets online.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

*This file is processed off-line.*

*It allows users to pay tickets online.*

*The system provides suggestions to architects.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

*This file is processed off-line.*

*It allows users to pay tickets online.*

*The system provides suggestions to architects.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

*This file is processed off-line.*

*It allows users to pay tickets online.*

*The system provides architects with suggestions.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

*This file is processed off-line.*

*It allows users to pay tickets online.*

*The system provides architects with suggestions.*

*I replaced the old rug by a new one.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

*This file is processed off-line.*

*It allows users to pay tickets online.*

*The system provides architects with suggestions.*

*I replaced the old rug by a new one.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

*This file is processed off-line.*

*It allows users to pay tickets online.*

*The system provides architects with suggestions.*

*I replaced the old rug with a new one.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

*This file is processed off-line.*

*It allows users to pay tickets online.*

*The system provides architects with suggestions.*

*I replaced the old rug with a new one.*

*Paper bags have been largely replaced with plastic bags.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

*This file is processed off-line.*

*It allows users to pay tickets online.*

*The system provides architects with suggestions.*

*I replaced the old rug with a new one.*

*Paper bags have been largely replaced with plastic bags.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

*This file is processed off-line.*

*It allows users to pay tickets online.*

*The system provides architects with suggestions.*

*I replaced the old rug with a new one.*

*Paper bags have been largely replaced by plastic bags.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

*This file is processed off-line.*

*It allows users to pay tickets online.*

*The system provides architects with suggestions.*

*I replaced the old rug with a new one.*

*Paper bags have been largely replaced by plastic bags.*

*We have evaluated more 2 systems.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

*This file is processed off-line.*

*It allows users to pay tickets online.*

*The system provides architects with suggestions.*

*I replaced the old rug with a new one.*

*Paper bags have been largely replaced by plastic bags.*

*We have evaluated more 2 systems.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

*This file is processed off-line.*

*It allows users to pay tickets online.*

*The system provides architects with suggestions.*

*I replaced the old rug with a new one.*

*Paper bags have been largely replaced by plastic bags.*

*We have evaluated two more systems.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

*This file is processed off-line.*

*It allows users to pay tickets online.*

*The system provides architects with suggestions.*

*I replaced the old rug with a new one.*

*Paper bags have been largely replaced by plastic bags.*

*We have evaluated two more systems.*

*Our approach detected 1.350 violations.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

*This file is processed off-line.*

*It allows users to pay tickets online.*

*The system provides architects with suggestions.*

*I replaced the old rug with a new one.*

*Paper bags have been largely replaced by plastic bags.*

*We have evaluated two more systems.*

*Our approach detected 1.350 violations.*

# Tip #23

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

*This file is processed off-line.*

*It allows users to pay tickets online.*

*The system provides architects with suggestions.*

*I replaced the old rug with a new one.*

*Paper bags have been largely replaced by plastic bags.*

*We have evaluated two more systems.*

*Our approach detected 1,350 violations.*

*The function is formalized as follows:*

*Each coefficient has a unique property that differs it from others.*

*This file is processed off-line.*

*It allows users to pay tickets online.*

*The system provides architects with suggestions.*

*I replaced the old rug with a new one.*

*Paper bags have been largely replaced by plastic bags.*

*We have evaluated two more systems.*

*Our approach detected 1,350 violations.*

**Tip #23: Common mistakes (16)**

- Avoid it!

# Tip #24

*ArchLint requires two inputs on the system under analysis: a high-level component specification and the history of revisions.*

# Tip #24

*ArchLint requires two inputs on the system under analysis: a high-level component specification and the history of revisions.*

# Tip #24

*ArchLint requires two inputs on the system under analysis: a high-level component specification and the history of revisions.*

*ArchLint requires two inputs on the system under analysis: (i) a high-level component specification and (ii) the history of revisions.*

# Tip #24

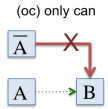
*ArchLint requires two inputs on the system under analysis: a high-level component specification and the history of revisions.*

*ArchLint requires two inputs on the system under analysis: (i) a high-level component specification and (ii) the history of revisions.*

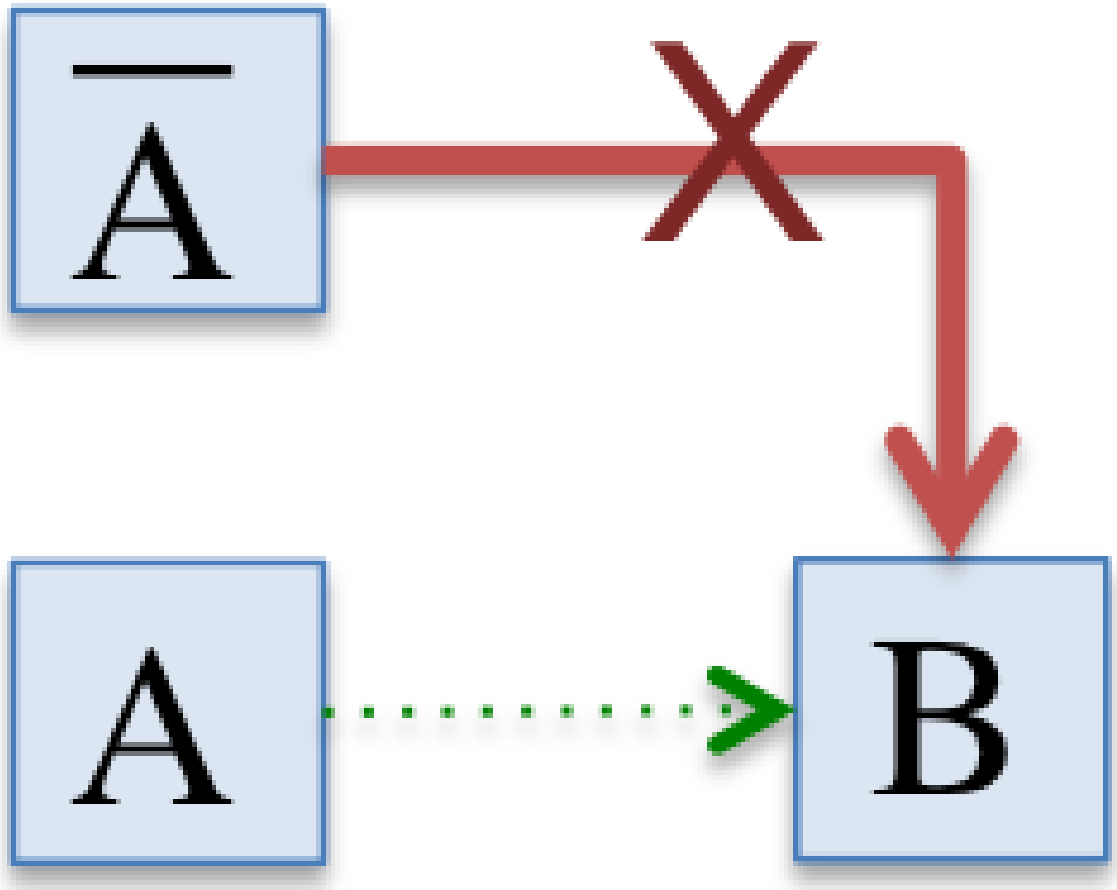
**Tip #24: Make itemsets clear**

- You can enumerate items throughout the text
  - However, it is better to make the items clear

# Tip #25

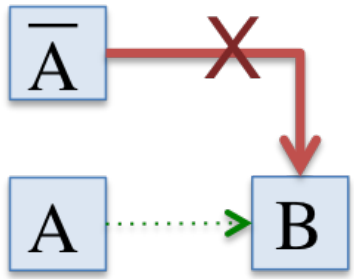


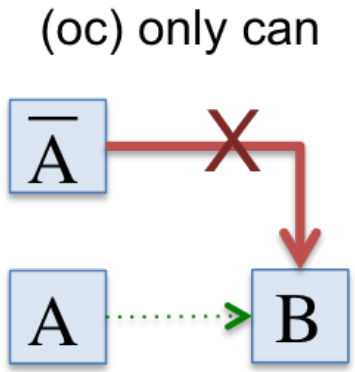
# (oc) only can



# Tip #25

(oc) only can






## Tip #25: Beware Sizes

- Figures, tables, etc. must have the appropriate size!
  - If small, "Do you want me to read?"
  - If big, "Do you need to take up space?"

# Tip #26

*LaTeX*

```
It was suggested to move class B to module B.
```

# Tip #26

*LaTeX*

```
It was suggested to move class B to module B.
```

# Tip #26

*LaTeX*

```
It was suggested to move class~B to module~B.
```

# Tip #26

*LaTeX*

```
It was suggested to move class~B to module~B.

... as described in section V.
```

# Tip #26

*LaTeX*

```
It was suggested to move class~B to module~B.

... as described in section V.
```

# Tip #26

*LaTeX*

```
It was suggested to move class~B to module~B.

... as described in Section~V.
```

# Tip #26

*LaTeX*

```
It was suggested to move class~B to module~B.

... as described in Section~V.

It has a heavy-user behavior w.r.t.  module~m.
```

# Tip #26

*LaTeX*

```
It was suggested to move class~B to module~B.

... as described in Section~V.

It has a heavy-user behavior w.r.t.  module~m.
```

# Tip #26

*LaTeX*

```
It was suggested to move class~B to module~B.

... as described in Section~V.

It has a heavy-user behavior w.r.t.~module~m.
```

*LaTeX*

```
It was suggested to move class~B to module~B.

... as described in Section~V.

It has a heavy-user behavior w.r.t.~module~m.

Our first evaluation relies on Jhotdraw.
```

# Tip #26

*LaTeX*

```
It was suggested to move class~B to module~B.

... as described in Section~V.

It has a heavy-user behavior w.r.t.~module~m.

Our first evaluation relies on Jhotdraw.
```

# Tip #26

*LaTeX*

```
It was suggested to move class~B to module~B.

... as described in Section~V.

It has a heavy-user behavior w.r.t.~module~m.

Our first evaluation relies on \mbox{JHotDraw}.
```

*LaTeX*

```
It was suggested to move class~B to module~B.

... as described in Section~V.

It has a heavy-user behavior w.r.t.~module~m.

Our first evaluation relies on \mbox{JHotDraw}.

Fowler et al. \cite{fowler99} proposed a
   catalog of refactorings.
```

*LaTeX*

```
It was suggested to move class~B to module~B.

... as described in Section~V.

It has a heavy-user behavior w.r.t.~module~m.

Our first evaluation relies on \mbox{JHotDraw}.

Fowler et al. \cite{fowler99} proposed a
   catalog of refactorings.
```

*LaTeX*

```
It was suggested to move class~B to module~B.

... as described in Section~V.

It has a heavy-user behavior w.r.t.~module~m.

Our first evaluation relies on \mbox{JHotDraw}.

Fowler et al.~proposed a catalog of
   refactorings~\cite{fowler99}.
```

*LaTeX*

```
It was suggested to move class~B to module~B.

... as described in Section~V.

It has a heavy-user behavior w.r.t.~module~m.

Our first evaluation relies on \mbox{JHotDraw}.

Fowler et al.~proposed a catalog of
   refactorings~\cite{fowler99}.

There are other architectural conformance
   tools~\cite{arch1}\cite{arch2}.
```

*LaTeX*

```
It was suggested to move class~B to module~B.

... as described in Section~V.

It has a heavy-user behavior w.r.t.~module~m.

Our first evaluation relies on \mbox{JHotDraw}.

Fowler et al.~proposed a catalog of
   refactorings~\cite{fowler99}.

There are other architectural conformance
   tools~\cite{arch1}\cite{arch2}.
```

*LaTeX*

```
It was suggested to move class~B to module~B.

... as described in Section~V.

It has a heavy-user behavior w.r.t.~module~m.

Our first evaluation relies on \mbox{JHotDraw}.

Fowler et al.~proposed a catalog of
   refactorings~\cite{fowler99}.

There are other architectural conformance
   tools~\cite{arch1,arch2}.
```

*LaTeX*

```
It was suggested to move class~B to module~B.

... as described in Section~V.

It has a heavy-user behavior w.r.t.~module~m.

Our first evaluation relies on \mbox{JHotDraw}.

Fowler et al.~proposed a catalog of
   refactorings~\cite{fowler99}.

There are other architectural conformance
   tools~\cite{arch1,arch2}.
```

**Tip #26: Avoid weird line breaks**

- To avoid undesirable line breaks, use tilde:
  - *before* identifiers (e.g., `B` and `V`) and cite command (`cite`)
  - *after* dots that are not the end of the sentence (e.g., `et al.` and `w.r.t.`)

## *BibTeX*

```
@inproceedings{doi21.312.123,
author = {Jonathan Aldrich, Craig Chambers, David Notkin},
title = {ArchJava:  connecting software architecture to implementation},
booktitle = {Proceedings of 22nd Int.  Conference on Software Engineering},
year = {2002},
pages = {187--197},
}
```

## *BibTeX*

```
@inproceedings{doi21.312.123,
author = {Jonathan Aldrich, Craig Chambers, David Notkin},
title = {ArchJava:  connecting software architecture to implementation},
booktitle = {Proceedings of 22nd Int.  Conference on Software Engineering},
year = {2002},
pages = {187--197},
}
```

## *BibTeX*

```
@inproceedings{doi21.312.123,
author = {Jonathan Aldrich, Craig Chambers, David Notkin},
title = {ArchJava:  connecting software architecture to implementation},
booktitle = {Proceedings of 22nd Int.  Conference on Software Engineering},
year = {2002},
pages = {187--197},
}
```

```
@inproceedings{archjava,
   author    = {Jonathan Aldrich and Craig Chambers and David Notkin},
   title     = {{ArchJava}:  connecting software architecture to implementation},
   booktitle = {22nd International Conference on Software Engineering (ICSE)},
   year      = {2002},
   pages     = {187-197},
}
```

## *BibTeX*

```
@inproceedings{doi21.312.123,
author = {Jonathan Aldrich, Craig Chambers, David Notkin},
title = {ArchJava:  connecting software architecture to implementation},
booktitle = {Proceedings of 22nd Int.  Conference on Software Engineering},
year = {2002},
pages = {187--197},
}
```

```
@inproceedings{archjava,
   author    = {Jonathan Aldrich and Craig Chambers and David Notkin},
   title     = {{ArchJava}:  connecting software architecture to implementation},
   booktitle = {22nd International Conference on Software Engineering (ICSE)},
   year      = {2002},
   pages     = {187-197},
}
```

### Tip #27: BibTeX

- Keep your BibTeX perfect
- Some guidelines:
  - Good organization (indentation), authors separated by "and", standardization (e.g., booktitle), capitalization of the title (e.g., {Java}), etc.

# Tip #28

We do **not** claim that our approach is complete. (emphasize)

# Tip #28

We do **not** claim that our approach is complete. (emphasize)

LaTeX: `{\bf not}` or `\textbf{not}`

# Tip #28

We do **not** claim that our approach is complete. (emphasize)

LaTeX: `{\bf not}` or `\textbf{not}`

This type of violation is known as *divergence*. (a very relevant word)

# Tip #28

We do **not** claim that our approach is complete. (emphasize)

LaTeX: `{\bf not}` or `\textbf{not}`

This type of violation is known as *divergence*. (a very relevant word)

LaTeX: `{\em divergence}` or `\textit{divergence}`

# Tip #28

We do **not** claim that our approach is complete. (emphasize)

LaTeX: `{\bf not}` or `\textbf{not}`

This type of violation is known as *divergence*. (a very relevant word)

LaTeX: `{\em divergence}` or `\textit{divergence}`

Class **`ArrayList`** is supertype of **`Collection`**. (identifiers)

# Tip #28

We do **not** claim that our approach is complete. (emphasize)

LaTeX: `{\bf not}` or `\textbf{not}`

This type of violation is known as *divergence*. (a very relevant word)

LaTeX: `{\em divergence}` or `\textit{divergence}`

Class **`ArrayList`** is supertype of **`Collection`**. (identifiers)

LaTeX: `\id{ArrayList}` where `id` is defined as follows:

```
\newcommand{\id}[1]{$\bm{\mathtt{#1}}$}
```

# Tip #28

We do **not** claim that our approach is complete. (emphasize)

LaTeX: `{\bf not}` or `\textbf{not}`

This type of violation is known as *divergence*. (a very relevant word)

LaTeX: `{\em divergence}` or `\textit{divergence}`

Class **`ArrayList`** is supertype of **`Collection`**. (identifiers)

LaTeX: `\id{ArrayList}` where `id` is defined as follows:

```
\newcommand{\id}[1]{$\bm{\mathtt{#1}}$}
```

Therefore, we consider target dependency $(c, t)$. (formulas)

# Tip #28

We do **not** claim that our approach is complete. (emphasize)

LaTeX: `{\bf not}` or `\textbf{not}`

This type of violation is known as *divergence*. (a very relevant word)

LaTeX: `{\em divergence}` or `\textit{divergence}`

Class **`ArrayList`** is supertype of **`Collection`**. (identifiers)

LaTeX: `\id{ArrayList}` where `id` is defined as follows:

```
\newcommand{\id}[1]{$\bm{\mathtt{#1}}$}
```

Therefore, we consider target dependency $(c, t)$. (formulas)

LaTeX: `$(c,t)$`

# Tip #28

We do **not** claim that our approach is complete. (emphasize)

LaTeX: `{\bf not}` or `\textbf{not}`

This type of violation is known as *divergence*. (a very relevant word)

LaTeX: `{\em divergence}` or `\textit{divergence}`

Class **`ArrayList`** is supertype of **`Collection`**. (identifiers)

LaTeX: `\id{ArrayList}` where `id` is defined as follows:

```
\newcommand{\id}[1]{$\bm{\mathtt{#1}}$}
```

Therefore, we consider target dependency $(c, t)$. (formulas)

LaTeX: `$(c,t)$`

**Tip #28: Writing Patterns**

- It is crucial to define a pattern when you write papers
- It can be any pattern, but **never** mixed them up

# Bibliography I


(1) *Can vs. May*.
http://data.grammarbook.com/blog/definitions/can-v-may/. 2007.

(2) CliffsNotes. *What's the difference between like and such as?*
http://www.cliffsnotes.com/cliffsnotes/writing/what-s-the-difference-between-like-and-such-as. 2013.

(3) *Comma with Because*.
http://languageandgrammar.com/2009/07/30/comma-with-because/. 2009.

(4) *Either/Or and Neither/Nor*. http://www.grammar-monster.com/lessons/either_or_neither_nor_double_negative.htm. 2013.

(5) Mignon Fogarty. *Which Versus That*.
http://www.quickanddirtytips.com/education/grammar/which-versus-that-0. 2008.

(6) *Forming Greek- and Latin-derived Nouns*.
http://web.utk.edu/~hirst/460/nouns.html. 2013.

(7) Maeve Maddox. *Cannot or Can Not?*
http://www.dailywritingtips.com/cannot-or-can-not/. 2009.

# Bibliography II


(8) Candice Chovanec Melzow. *When to Use Former and Latter*. http://www.uhv.edu/ac/newsletters/writing/grammartip2007.10.31.htm. 2007.

(9) Merriam-Webster. *English Grammar*. http://www.merriam-webster.com/. 2013.

(10) Saint Mary's University of Minnesota. *Phrasal Verbs*. http://www2.smumn.edu/deptpages/tcwritingcenter/otherresources/grammar_punctuation_files/PhrasalVerbs.pdf. 2013.

(11) Vânia Maria do Nascimento Duarte. *Uso do etc.* http://www.portugues.com.br/gramatica/uso-etc-.html. 2013.

(12) The Writing Center of The University of North Carolina. *Passive Voice*. http://writingcenter.unc.edu/handouts/passive-voice/. 2013.

(13) Geoff Pope. *"Like" Versus "Such As"*. http://www.quickanddirtytips.com/education/grammar/like-versus-such-as. 2013.

(14) *Rules for Comma Usage*. http://grammar.ccc.commnet.edu/grammar/commas.htm. 2013.

(15) *Spelling*. http://www.writers.com/tips_spelling.html. 2013.

# Bibliography III


(16) John Swales and Christine B. Feak. *Academic writing for graduate students: essential tasks and skills*. 3rd ed. University of Michigan Press/ELT, 2012.

(17) Vanessa Tavares. *What's the difference between OTHER, OTHERS and ANOTHER?* http://english4youonline.blogspot.com.br/2011/03/whats-difference-between-other-others.html. 2011.

(18) *The Comma: Major Uses and Worst Abuses*. http://wire.rutgers.edu/p_grammar_comma2.html. 2013.

(19) April Toadvine et al. *Active and Passive Voice*. https://owl.english.purdue.edu/owl/owlprint/539/. 2013.

(20) University of Victoria English Language Centre. *Using Commas*. http://web2.uvcs.uvic.ca/elc/studyzone/330/grammar/comma.htm. 2013.

(21) Get It Write. *Which or That?* http://www.getitwriteonline.com/archive/103103whichthat.htm. 2003.

Thanks!!!