

CSCI 3155: Principles of Programming Languages
Exercise sheet #1
4th June 2007

Name: _____

Language Feature Evaluation

You must hand in today's exercise by yourself, though you may collaborate with others to find suitable answers. Tomorrow, you may hand in your homework together with up to two other students, or by yourself. Starting wednesday, you must have formed a group of two or three. If you find that you cannot form a group with other students, contact the instructor as soon as possible.

Today's exercise is based on Chapter 1 in the book, particularly on Table 1.1 and its accompanying discussion. Since some of you may not yet have the textbook, the text below includes some information that will help you with the exercises.

The three criteria by which we evaluate language features are READABILITY, WRITABILITY, and RELIABILITY. The ten language characteristics that we discussed earlier in class were Simplicity, Orthogonality, Control structures, Data types and structures and classes, Syntax design, Support for abstraction, Expressivity, Type checking, Exception and error handling, and Restricted aliasing. If you believe that you do not sufficiently understand one of the criteria or characteristics, do not hesitate to ask the instructor.

Exercise 1. The object-oriented general-purpose programming language *Ruby* provides two means for constructing lists of strings, illustrated below:

```
list = [ 'foo', 'bar', 'quux' ]  
list = %w{ foo bar quux }
```

The two statements above assign the same list of strings to `list`.

- (a) (**Skills 1.1, 1.2**) Which of our ten characteristics are affected by this feature, and why?
- (b) (**Skill 1.3**) How does this feature affect readability, writability and reliability?

Exercise 2. Most programming languages overload the addition operator “+” to support both integer and floating point addition. A notable exception is the general-purpose programming language *OCaml*, which uses “+” for integer addition and “+.” for floating-point addition.

- (a) (**Skills 1.1, 1.2**) Which of our ten characteristics are affected by this choice, and why?
- (b) (**Skill 1.3**) How does this design choice affect readability, writability and reliability?

Exercise 3. (Book, p36, Problem Set #1) What are some features of specific programming languages you know whose rationale escapes you?

Exercise 4. The programming language *awk* implicitly performs string concatenation on all expressions that are placed next to each other. For example, the program fragment

```
i = 1;
result = "hello";
print ("#" i ":" result "world!");
```

will print “#1: hello world!”. Most other programming languages use an explicit concatenation operator, such as `+`, `++`, `.` or `^`.

- (a) (**Skill 1.4**) Do you agree with the choice of the *awk* language designers? Argue using our criteria.
- (b) (**Skill 1.4**) *awk* was designed as a line-based string processing language. How does this information affect your previous critique?

Exercise 5. You are asked to design a new general-purpose programming language.

- (a) (Book, p36, Problem Set #12) What language features will you insist on putting into the language, and why?
- (b) (**Skill 1.1**) Pick two characteristics you will emphasise in your design, and two that you will de-emphasise. Explain your choices.