

Programming Languages G22.2110 Summer 2007 hw02

Assigned Th 5/24/2007, due We 5/30/2007 at 1pm.

How to Submit Homework Assignments

Email your answers, in either plain text format or as pdf, to Abhijit Guria <guria@cs.nyu.edu>. Assignments are due on Wednesdays at 1pm. This deadline will be strictly enforced.

Reading Assignments

- For lecture on 5/24/2007: Scott 6.0, 6.1.0-6.1.1;
van Rossum 1-5 (<http://docs.python.org/tut/tut.html>)
 - For lecture on 5/31/2007: Scott 10.0, 10.2, 10.3.0-10.3.4;
Candolin (<http://www.cs.hut.fi/Studies/T-93.210/schemetutorial/>)
-

Homework Assignments

1. Syntax (10 = 5 + 5 points)
Consider the following grammar:

```
(start symbol EXPR)
EXPR ::= EXPR OP EXPR | NUM
OP    ::= ':=' | '==' | '!='
NUM   ::= '0' | '1' | '2' | '3'
```

- 1a. (5 points) Write a grammar that recognizes the same language, but gives := a lower precedence than ==, and gives == the same precedence as !=.
- 1b. (5 points) Write a grammar that recognizes the same language, enforces the same precedence rules as in Question 1a, makes := right-associative, and makes both == and != left-associative.

2. Python (8 points)

While you write the Python code for answering Question 3 below, you will probably get some error messages. Describe two error messages using the following format:

- Code: *a very short piece of code that triggers the error*
- Symptom: *the error message itself*
- Cause: *an explanation for what triggered the error message*
- Solution: *how to fix the code to prevent the error*

3. Python (20 = 5 + 5 + 5 + 5 points)

Write Python programs exercising the fundamental features.

3a. I/O (5 points)

Write a program that prompts the user for his or her name, reads the name from input, then politely greets the user by name. Here is an example interactive session:

```
What is your name?  
Bob  
Hello, Bob, nice to meet you!
```

3b. Libraries (5 points)

Write a program that uses Python library functions to compute $\sqrt{2}$, $\sin(3.5)$, and $e^{2.5}$, and then prints the results like this (don't worry if the decimals aren't 100% identical):

```
square root of 2.0:    1.414214  
sine of 3.5:         -0.350783  
e to the power of 2.5: 12.182494
```

3c. Types (5 points)

The following code creates a variable `b` with the boolean value `True`, and then prints a description and the value of the variable:

```
b = True  
print 'name b, type boolean, value %s' % b
```

Extend this program by creating and printing more variables of different types. Your program should produce the following output:

```
name b, type boolean, value True  
name d, type dictionary, value {'kiwi': 'green', 'apple': 'red'}  
name f, type float, value 3.141  
name i, type int, value 42  
name l, type list, value [1, 4, 9, 16]  
name s, type string, value hello  
name t, type tuple, value ('answer', 42)
```

3d. Control flow (5 points)

Write a Python function `count_occurrences` that takes two parameters, a string and a character, and returns the number of occurrences of the character in the string. For example, `count_occurrences('hello', 'l')` should return 2.

4. Scheme (0 points)

Start teaching yourself Scheme by doing the following:

- 4a. If possible, find peers (other students who want to learn Scheme together with you) and gurus (people who already know Scheme, whom you can ask questions when you get stuck).
- 4b. Make sure you have access to a Scheme interpreter in your preferred working environment. One possibility is the `guile` interpreter, which is installed on most GNU systems. Otherwise, a good choice is DrScheme. If you use DrScheme, you will have to pick a “language level”; choose the Standard (R5RS).
- 4c. Read the Scheme tutorial (see reading assignment above). Along the way, try things out with the Scheme interpreter that you installed in Step b.
- 4d. Find an online version of the R5RS, and familiarize yourself with its table of contents. In particular, find the chapter describing common functions that are available in all implementations that conform to the standard, such as `guile` or DrScheme.

5. Scheme (12 = 3 + 3 + 3 + 3 points)

Continue teaching yourself Scheme by reading the example code from

<http://www.cs.nyu.edu/courses/summer07/G22.2110-001/hw02-scm-example.txt>

- 5a. (3 points) What does the program print?
- 5b. (3 points) Add the line `(display (toString '(MUL (CONST 3) (VAR x))))` to the bottom of the script. Based on the output, describe briefly what function `toString` does.
- 5c. (3 points) Add the line `(display "op == ") (display op) (newline)` right before the `(case ...)` expression in function `toString`. Add similar debugging output for `s1` and `s2`. Based on your experiments, explain briefly what the `let` binding does.
- 5d. (3 points) Use debugging output to figure out what function `diff` does. Describe briefly what each clause of the case-expression in `diff` implements.

<http://www.cs.nyu.edu/courses/summer07/G22.2110-001/hw02.pdf>

Total points: 50.