Python Project
Due Wednesday, 6/1/2011

This is your chance to apply what we have learned this term to analyze a fairly recent language that
has become quite popular: Python. (In fact, we will be teaching it in our first CS course starting next
fall.) Its designers claim that it is a multi-paradigm language. You will be asked to examine that claim.

Read about Python. There are lots of tutorials and books out there; find one or more that appeal
to you. (I am choosing not to recommend any particular one.) Then answer the following questions.
Each can be answered in at most a few paragraphs. Please type your answers.

1. (15 points) Overview
   Give a basic overview of the language. What interesting or unusual features does it have? Include
   such things as its type system, scoping rules, how its programs are structured, its parameter
   passing conventions, built-in data types, etc.

2. (10 points) Python as an imperative language
   How well does Python fulfill the expectations for an imperative language? How easily could a
typical C program be translated into Python?

3. (15 points) Python as a functional language
   How easy is it to program in a functional style in Python? What support does it give for this
   style? Are functions first-class objects? Is there support for higher-order functions? Could you
easily translate Parser68.hs into Python? Why or why not?

4. (15 points) Python as an Object Oriented language
   How well does Python function as an object-oriented language? Does it support data abstraction
   well? Inheritance? Subtyping? Could the Java Collections class be translated easily into Python?
   Why or why not?

5. (10 points) Purpose and Evaluation
   What was the purpose for which Python was designed? Which features does it include or leave
   out because of this purpose? Does it seem to fulfill its purpose well? Are there purposes for which
   it would be poorly suited? Why?

6. (0 points) Bibliography
   What references did you use to learn about Python and answer these questions? Which ones did
   you find particularly useful? (I am giving 0 points because there are no right or wrong answers,
   but you do need to specify your sources!)