1 Administrative Topics

- We take the quiz.

2 Computing the Student GPA

Our next goal is to compute the GPA, and to do that, it would be nice to have a list of numbers instead of a list of duples. So, let’s make a method that extracts the number grades from the duples list:

```python
# return a list of numeric grades
def getNumericGrades(self):
    nums = []
    for duple in self.grades:
        nums.append(duple[1])
    return nums
```

Finally, we are ready to write `getGPA`. The most interesting aspect of this function is the syntax for calling one method from another. We use the dot notation with the `self` variable as the object.

```python
# return the average of the number grades
def getGPA(self):
    ng = self.getNumericGrades()
    if len(ng) == 0:
        return 0.0
```

avg = 0.0
for g in ng:
    avg += g  # equiv to avg = avg + g
avg /= len(ng)  # equiv to avg = avg / len(ng)
return avg

The complete code has been uploaded to the course web page.
3 Miscellany

Here is a summary of the Student class.
It has data fields

- name
- year
- grades

and methods

- getName (an accessor method)
- getYear (an accessor method)
- setYear (a mutator method)
- __init__
- addGrade (a mutator method)
- getGrades (an accessor method)
- getNumericGrades (an accessor method)
- getGPA (an accessor method)

And here are a couple of notes:

- Note that there is an entry for each data field and each method in a Student object’s symbol table. One implication is that it means you cannot make methods and data fields with the same name.

- Also, note that the methods are added to the object’s symbol table when Python first creates the object. The data fields are added when __init__ is executed. As a general rule, if any data field that is ever going to be in an object, it should be added in the __init__ method. This makes for easy reading later.
4 Advice for Project 7

We may have time to discuss the lsystem and interpreter class test function.