CSE 453

Inheritance

What is Inheritance?

- The creation of a new class from an existing class
- The new class inherits and then extends the members of class it inherits

Terminology

- Superclass
  - The general (inherited) class
  - *aka*, base class
- Subclass
  - The specialized (extended) class
  - *aka*, derived class

Polymorphism

- Allows subclasses to have methods with the same names as methods in their superclasses
- Gives program the ability to call the correct method depending on the type of object that is used to call it
- When a method is redefined in the subclass it overrides the method in the superclass

The isinstance Function

- The `isinstance` function checks to determine if an object is of a particular class
- Format

  `isinstance(`*`object`*`, `*`ClassName`*`)`
  - *object* is a variable that references an object
  - *ClassName* is the name of the class the object is being checked against
- Returns True if *object* belongs to the class *ClassName*
  - Otherwise returns False

Example

```
class Student:
        def __init__(self,pupil,midterm,final):
                self.__student = pupil
                self.__midterm = midterm
                self.__final = final
                self.__avg = 0.0
                self.__grade = ''

        def set_midterm_exam(self,score):
                self.__midterm = score
        def set_final_exam(self,score):
                self.__final = score
```

```
        def get_midterm_exam(self):
                return self.__midterm
        def get_final_exam(self):
                return self.__final
        def get_student(self):
                return self.__student
        def __str__(self):
                self.__avg=(float(self.__midterm)+float(self.__final))/2
                if self.__avg < 65.0:
                        self.__grade = 'failing'
                else:
                        self.__grade = 'passing'
                return 'The '+ self.__student + ' is ' + self.__grade + \
                       ' with an average of ' + str(self.__avg)

class CSE408_Student(Student):
        def __init__(self, name, midterm, final, proj, username):
                Student.__init__(self, name, midterm, final)
                self.__project=proj
                self.__avg=0.0
                self.__grade=''
        def set_project(self, proj):
                self.__project=proj
        def __str__(self):
                self.__avg = (float(self.get_midterm_exam()) + \
                        float(self.get_final_exam()) +
float(self.__project))/3
                if self.__avg < 65.0:
                        self.__grade = 'failing'
                else:
                        self.__grade = 'passing'
                return 'The ' + self.get_student() + ' is ' + \
                   self.__grade + ' with an average of ' \
                   + str(self.__avg)

mary = CSE408_Student('Mary Jones',0,0,0,'mj')
mary.set_final_exam(84)
mary.set_midterm_exam(61)
mary.set_project(78)
print mary
```

Creating a Class That Inherits Another Class

- Consider the previous example
  - ☞ Notice the first line of the definition of the class **`CSE408_Student`** (subclass) uses **`Student`** (superclass) as an argument:
    `class CSE408_Student(Student):`
  - ☞ Notice the third line of the definition of the class **`CSE408_Student`** (subclass) uses **`Student.__init__`** to inherit its properties:
    `Student.__init__(self, name, midterm, final)`

References

- Tony Gaddis, *Starting Out With Python*, Second Edition, Pearson Education, Inc. (Addison Wesley), 2012