

**ELEC 279 - Winter 2020**  
Introduction to Object-Oriented Programming  
**Lab 3 - Week 6**  
Inheritance

**Introduction:** In object-oriented programming, inheritance is the process that allows a class to inherit the properties (methods and attributes (instance variables)) of another class. Inheritance makes the information appear in a hierarchical order. There are two essential class definitions in the context of inheritance. The **subclass** (also called derived or child class) inherits the attributes (instance variables) and methods of another class; the class whose properties are inherited is the **superclass** (also called base or parent class).

Inheritance allows a programmer to separate methods and attributes that are shared between subclasses and those that are unique to each particular subclass. It also allows subclasses to **override** or **overload** some inherited methods. In Java, although multiple inheritance (a subclass can have more than one superclass) is not supported, multilevel inheritance, in which a subclass can pass its properties to subclasses, is supported. Figure 1 depicts a single inheritance hierarchy where three subclasses inherit the properties of one superclass, although having their sets of unique properties.

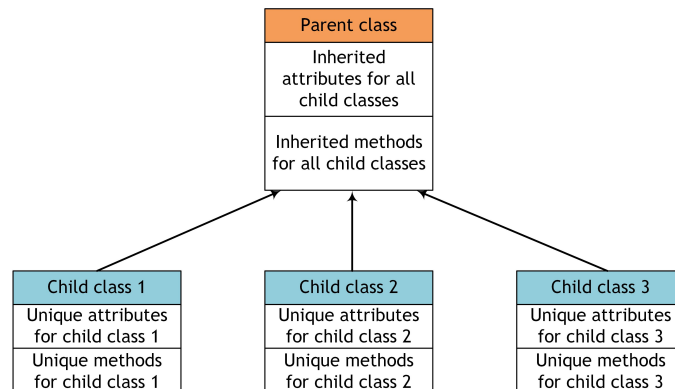


Figure 1: Single inheritance.

In java, the keyword ***extends*** is used to define a subclass. This example shows how to have the class ***Instructor*** to inherit from the class ***Staff***.

```
public class Staff {
    public String position;
    public void Salary() {
        System.out.println("Staff takes a monthly salary.");
    }
}
```

```
public class Instructor extends Staff {
    public String officeHour;
    Instructor() {
        position = "Teaching";
    }
}
```

As you can see, the **Instructor** class inherits the *position* variable from the **Staff** class and has the unique *officeHour* variable. We can now declare an *Instructor* object, define the *officeHour* attribute, and call the *Salary* method and *position* attribute of their superclass:

```
class MyClass {
    public static void main(String [] args) {
        Instructor I = new Instructor();
        I.officeHour = "Monday, 3 pm";
        I.Salary();
        System.out.println(I.position);}
}
```

**Lab 3 Goals:** At the end of this Lab session (Lab 3), you would have accomplished the following:

1. Implement different tasks that cover basic inheritance concepts.
2. Understand the concept of methods overriding and overloading through the manipulation of pre-defined superclass methods.
3. Implement multi-level inheritance.

### LAB 3 - TASKS

**Problem:** Suppose we want to have an application that models students in an educational system. A parent class **Person** holds common attributes such as name, address, and ID number, whereas child class; **Student** contains some unique properties for students. For students, the courses taken and their grades need to be maintained; add a course with grade and print all courses taken. Assume that the maximum number of courses a student can take throughout the entire program is 20. Moreover, a subclass **GradStudent** is a child class of **Student** and inherits the properties of both **Person** and **Student** classes. It also contains some unique properties for graduate students. For graduate students, the two comprehensive exams and thesis defense exam results need to be stored. Moreover, the number of committee members who attended these exams needs to be maintained.

#### Task 1: Creating the sub-class **Student**.

1. Create a **Lab3** project on Eclipse.
2. Download *Person.java* from onQ under **Lab 3 - Week 6** and copy it to your Lab 3 src folder.
3. Create a public sub-class **Student** that inherits the properties of **Person** class. This class should define a private integer attribute to keep track of the number of taken courses, a private string list to store the course codes, a private integer list to store the grades, and a private integer attribute that is of type **static** and can not be overridden to store the maximum number of courses a student can take. The maximum number of courses should be 20.

4. Inside this sub-class the following methods should be implemented:

- (a) A public **constructor** that receives the name, address, and ID number. These attributes are already defined in the base class **Person** so you do not need to redefine them in **Student**. You just need to invoke the super constructor to initialize them. In addition, you should initialize all the other instance variables defined in **Student**.
- (b) The public method **toString** overrides the superclass method and returns the string "Student: " prior to output of the superclass method.
- (c) The public method **addGrade** receives a string (representing the course code) and an integer (representing the course grade). The course code and grade should be added to the corresponding previously defined lists and the number of courses taken should be updated.
- (d) The public method **printGrades** that prints all of the taken courses and their corresponding grades.

5. In the main method, you can test your code with the following initialization:

```
public class Test {  
    /* Test Student class */  
    public static void main(String[] args) {  
        Person p1 = new Person("Julia", "181 University Ave.", 121456);  
        System.out.println(p1.toString());  
        Student s1 = new Student("John", "189 Univerity Ave.", 246718);  
        s1.addGrade("ELEC_279", 97);  
        s1.addGrade("ELEC_221", 68);  
        System.out.printf(s1.toString());  
        s1.printGrades();  
    }  
}
```

Your output should look similar to this:

```
Julia: 121456 (181 University Ave.)  
Student: John: 246718 (189 Univerity Ave.)  
ELEC 279:97  
ELEC 221:68
```

**CHECKPOINT:** Show the outputs to the Grad. TA for credit. It is time to switch driver and navigator roles (2 Marks).

**Task 2: Creating a subclass *GradStudent*:** In this task, it is required to maintain the results of the two comprehensive exams as well as the thesis defense exam for graduate students. Since results of these exams are either pass or fail, the *addGrade* method will be overloaded. Its input arguments will be two string variables that correspond to an exam's name and result. The student can fail the exam only once, which makes the maximum number of exam trials = 6

1. Create a public sub-class **GradStudent** that inherits the properties of **Student**. Inside this class, you should define a private integer for the number of committee members, a private integer to track the number of taken exams, a private string list that contains the exams' names and a private string list that maintains the grades of these exams. A private static integer for the maximum number of exams' trials should also be defined (let it be 6). Define it so that it can not be overridden.
2. Inside this sub-class the following methods should be implemented:
  - (a) A public **constructor** that receives the name, address, and ID number. Again, these attributes are inherited from the base class **Student**, which are already inherited from **Person**, so you do not need to redefine them. You just need to invoke the super constructor to initialize them. In addition, you should initialize all the other instance variables defined in **GradStudent**.
  - (b) The public methods **setComitteeNum** and **getComitteeNum** which are the mutator and accessor methods for the number of committee members.
  - (c) The public method **addGrade** that receives the name and grade of an exam and then stores the results in *Exam* and *Result* lists. This method overloads the original **addGrade** method that was previously defined in **Student** superclass.
  - (d) The public method **toString** overrides the superclass **Student** method and returns the string "Grad\_" prior to output of the superclass method.
  - (e) The public method **printExamGrades**. In this method, all names and grades of the exams taken by the graduate student are printed.
3. In the main method, you can test your code with the following initialization:

```
public class Test {
    public static void main ( String [] args ) {
        Person p1 = new Person("Julia", "181_University_Ave.", 121456);
        System.out.println(p1.toString());
        Student s1 = new Student("John", "89_Univerity_Ave.", 24671);
        s1.addGrade("ELEC_279", 97);
        s1.addGrade("ELEC_221", 68 );
        System.out.printf(s1.toString());
        s1.printGrades();
        Grad Student s2 = new Grad Student("Tom", "92_Reg._st.", 1014);
        s2.addGrade("ELEC_279", 97);
        s2.addGrade("ELEC_271", 88);
        s2.setComitteeNum(6);
        s2.addGrade("CompI", "PASS");
        s2.addGrade("COMPII", "PASS");
        s2.addGrade("Thesis_Defense_first_trial", "FAIL");
        s2.addGrade("Thesis_Defense_second_trial", "FAIL");
        s2.printExamGrades();
        s2.printGrades();}
}
```

Your output should look similar to this:

Julia: 121456 (181 University Ave.)  
Student: John: 246718 (89 Univerity Ave.)  
ELEC 279:97  
ELEC 221:68  
Grad Student: Tom: 1014 (92 Reg. st.)  
Compl :PASS  
COMPII :PASS  
Thesis Defense first trial: FAIL  
Thesis Defense second trial: FAIL  
ELEC 279:97  
ELEC 271:88

**CHECKPOINT:** Each group member should demonstrate their result to Grad TA for credit (1 Mark).

**End of Lab 3**