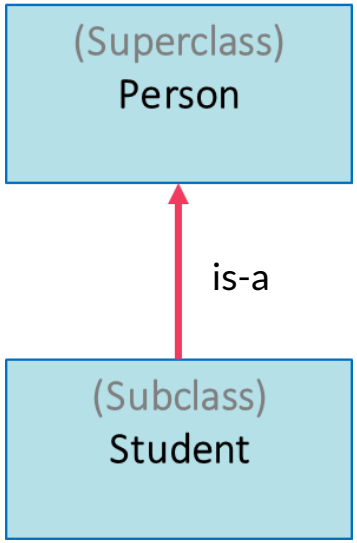
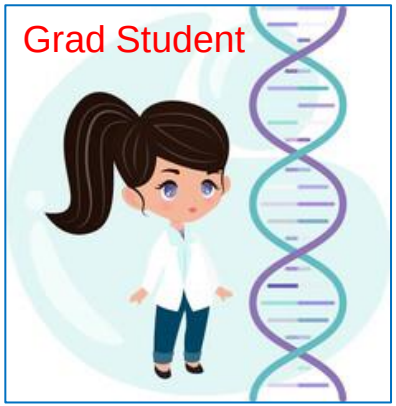
Java Inheritance



- Inheritance allows a class (known as the **subclass** or **derived** class) to acquire *properties* and *behaviors* from another class (known as the **superclass** or **base** class).

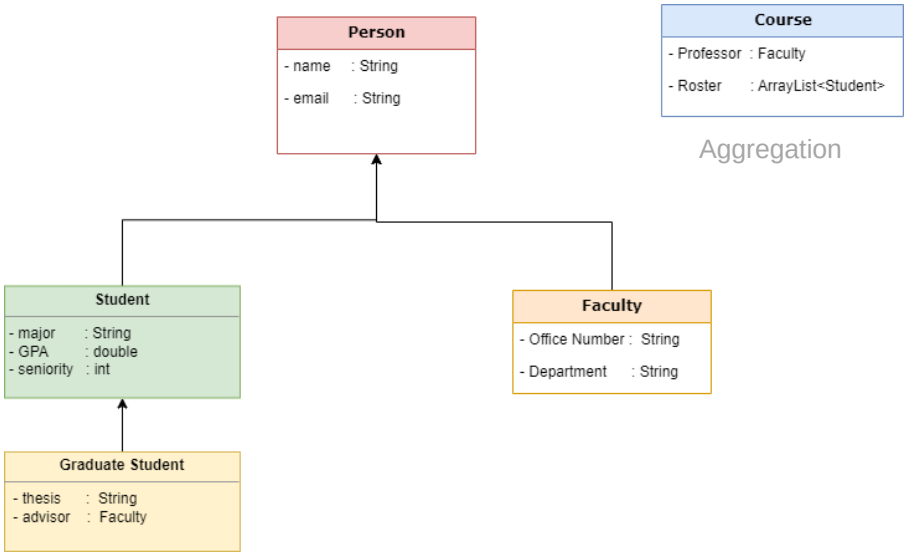
- Inheritance establishes an "**is-a**" relationship between classes, where the subclass is a specialized version of the superclass.

The College Case



The Person Hierarchy Case

Using Java classes (inheritance & aggregation) to model a typical University environment.



Person Class

Person Class
- name : String - email : String
+ Mutators + Accessors + Constructors + User-Defined Methods (toString titleCaps,...)

Student Class

Student : Person
- major : String - GPA : double
+ Mutators + Accessors + Constructors + User-Defined Methods (toString)

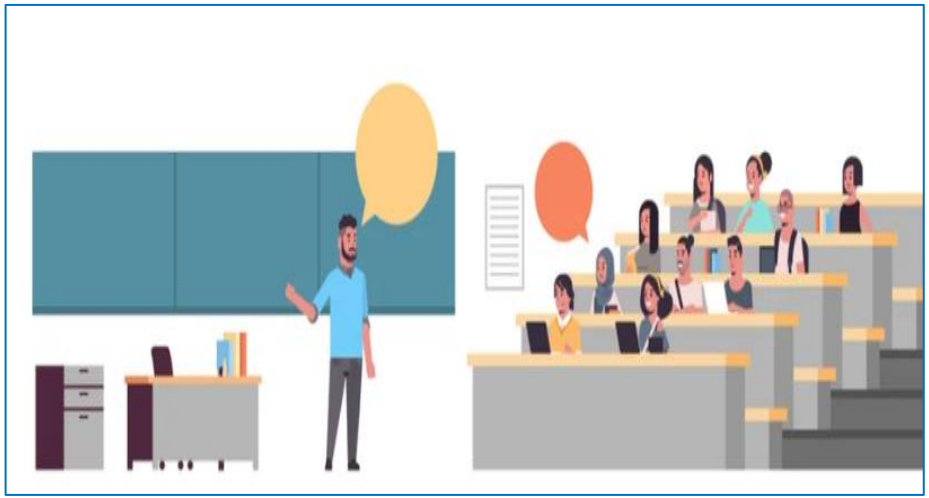
Graduate Student Class

GradStudent : Student
- thesis : String - advisor : String
+ Mutators + Accessors + Constructors + User-Defined Methods (toString)

Faculty Class

Faculty : Person
- office# : String - department : String
+ Mutators + Accessors + Constructors + User-Defined Methods (toString)

Course Class



Course (Aggregation)
- faculty : Faculty - roster : ArrayList<Student>
+ Mutators + Accessors + Constructors + User-Defined Methods (toString)

Java Code

```java
class Person {
    //Data members (Good practice: provide default values)
    private String name  = "na";
    private String email = "none";

    //Constructor (all-arguments, zero-args uses the default values)
    public Person(String nameVal, String emailVal) {
        setName(nameVal);
        setEmail(emailVal);
    }

    //Mutators (Put validation logic, business-rules here)
    public void setName(String nameVal)   { this.name = nameVal; }
    public void setEmail(String emailVal) { this.email = emailVal; }

    //Accessors
    public String getName()  { return name; }
    public String getEmail() { return email; }

    //User-defined methods
    @Override
    public String toString() {
        return "Pers. Name: " + name + "\nPers. Email: " + email;
    }
}
```

```java
class Student extends Person {
    private String major = "undeclared";
    private int    seniorityLevel = 1;
    private double gpa = 0;

    public Student(String name, String email,
                   String major, int seniorityLevel, double gpa) {
        super(name, email);
        setMajor(major);
        setSeniorityLevel(seniorityLevel);
        setGpa(gpa);
    }

    public void setMajor(String major) { this.major = major; }
    public void setGpa(double gpa) { this.gpa = gpa; }
    public void setSeniorityLevel(int seniorityLevel) {
        this.seniorityLevel = seniorityLevel;
    }

    public String getMajor() { return major; }
    public int    getSeniorityLevel() { return seniorityLevel; }
    public double getGpa() { return gpa; }



    @Override
    public String toString() {
        return super.toString() + "\nStud. Major: " + major
                                + "\nStud. Seniority Level: " + seniorityLevel
                                + "\nStud. GPA: " + gpa;
    }
}
```

Java Code

```java
class GradStudent extends Student {
    private String thesis = "na";
    private String advisor = null;

    public GradStudent(String name, String email,
                       String major, int seniorityLevel, double gpa,
                       String thesis, String advisor) {
        super(name, email, major, seniorityLevel, gpa);
        setThesis(thesis);
        setAdvisor(advisor);
    }

    public void setThesis(String thesis)   { this.thesis = thesis; }
    public void setAdvisor(String advisor) { this.advisor = advisor; }

    public String getThesis()  { return thesis; }
    public String getAdvisor() { return advisor; }

    @Override
    public String toString() {
        return super.toString() + "\nGrad. Thesis: " + thesis
                                + "\nGrad. Advisor: " + advisor;
    }
}
```

```java
class Faculty extends Person {
    private String officeNumber;
    private String department;

    public Faculty(String name, String email,
                   String officeNumber, String department) {
        super(name, email);
        setOfficeNumber(officeNumber);
        setDepartment(department);
    }

    public void   setOfficeNumber(String officeNumber) { this.officeNumber = officeNumber; }
    public String getDepartment() { return department; }

    public String getOfficeNumber() { return officeNumber; }
    public void   setDepartment(String department) { this.department = department; }

    @Override
    public String toString() {
        return super.toString() + "\nFac.  Office Number: " + officeNumber
                                + "\nFac.  Department: " + department;
    }
}
```

Java Code

```java
class Course {
    private Faculty faculty;
    private ArrayList<Student> roster;

    public Course(Faculty faculty, ArrayList<Student> roster) {
        setFaculty(faculty);
        setRoster(roster);
    }


    public void setFaculty(Faculty faculty) {
                        this.faculty = faculty; }
    public void setRoster(ArrayList<Student> roster) {
                        this.roster = roster; }

    public Faculty getFaculty() { return faculty; }
    public ArrayList<Student> getRoster() { return roster; }

    @Override
    public String toString() {
        StringBuilder sb = new StringBuilder();
        sb.append("Course-Faculty Information:\n")
                  .append(faculty)
                  .append("\n\n");
        sb.append("Course-Student Roster:\n");

        for (Student student : roster) {
            sb.append(student).append("\n\n");
        }
        return sb.toString();
    }
}
```

```java
public class Driver {
    public static void main(String[] args) {

        Person person = new Person("Juan Valdez", "greatCoffee@gmail.com");
        System.out.println("\n" + person);

        Student student = new Student("Harry Potter", "magic@gmail.com",
                                      "Magical Science", 4, 3.79);
        System.out.println("\n" + student);

        GradStudent gradStudent = new GradStudent("Sheldon Cooper", "sheldon@gmail.com",
                                                  "Astrophysics", 6, 3.99,
                                                  "Big-Bang Theory", "Dr. Xavier");
        System.out.println("\n" + gradStudent);

        Faculty faculty = new Faculty("Dr. Charles Xavier", "xman@gmail.com",
                                      "A123", "Physics");
        System.out.println("\n" + faculty);

        ArrayList<Student> courseRoster = new ArrayList<>();
        courseRoster.add(student);
        courseRoster.add(gradStudent);
        courseRoster.add(new Student("Hermione Granger", "her@gmail.com",
                                     "Mathematics", 4, 4.0));

        Course course = new Course(faculty, courseRoster);
        System.out.println("\n\nCourse Information----------------");
        System.out.println(course);

    }//main

}//Driver class
```

Output

```
Pers. Name: Juan Valdez
Pers. Email: greatCoffee@gmail.com

Pers. Name: Harry Potter
Pers. Email: magic@gmail.com
Stud. Major: Magical Science
Stud. Seniority Level: 4
Stud. GPA: 3.79

Pers. Name: Sheldon Cooper
Pers. Email: sheldon@gmail.com
Stud. Major: Astrophysics
Stud. Seniority Level: 6
Stud. GPA: 3.99
Grad. Thesis: Big-Bang Theory
Grad. Advisor: Dr. Xavier

Pers. Name: Dr. Charles Xavier
Pers. Email: xman@gmail.com
Fac.  Office Number: A123
Fac.  Department: Physics
```

```
Course Information----------
Course-Faculty Information:
Pers. Name: Dr. Charles Xavier
Pers. Email: xman@gmail.com
Fac.  Office Number: A123
Fac.  Department: Physics

Course-Student Roster:
Pers. Name: Harry Potter
Pers. Email: magic@gmail.com
Stud. Major: Magical Science
Stud. Seniority Level: 4
Stud. GPA: 3.79

Pers. Name: Sheldon Cooper
Pers. Email: sheldon@gmail.com
Stud. Major: Astrophysics
Stud. Seniority Level: 6
Stud. GPA: 3.99
Grad. Thesis: Big-Bang Theory
Grad. Advisor: Dr. Xavier

Pers. Name: Hermione Granger
Pers. Email: her@gmail.com
Stud. Major: Mathematics
Stud. Seniority Level: 4
Stud. GPA: 4.0
```