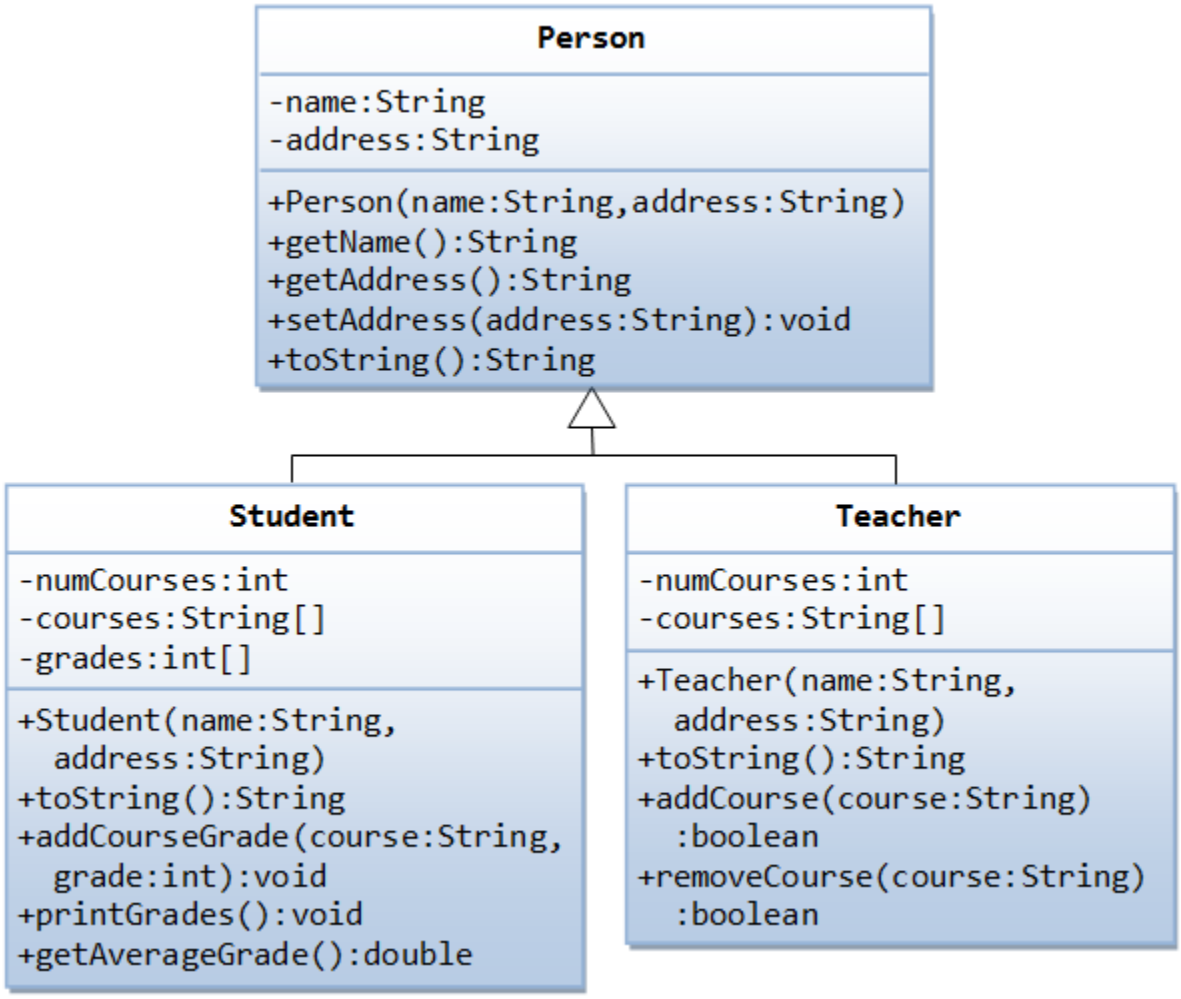
Lab 5

Example: `Person` and its subclasses



Suppose that we are required to model students and teachers in our application. We can define a superclass called `Person` to store common properties such as `name` and `address`, and subclasses `Student` and `Teacher` for their specific properties. For students, we need to maintain the courses taken and their respective grades; add a course with grade, print all courses taken and the average grade. A student takes no more than 30 courses for the entire program. For teachers, we need to maintain the courses taught currently, and able to add or remove a course taught. A teacher teaches not more than 5 courses concurrently.

We design the classes as follows.

Superclass `Person.java`

```
// Define superclass Person
public class Person {
   // Instance variables
   private String name;
   private String address;

   // Constructor
   public Person(String name, String address) {
      this.name = name;
      this.address = address;
   }

   // Getters
   public String getName() {
      return name;
   }
   public String getAddress() {
      return address;
   }

   public String toString() {
      return name + "(" + address + ")";
   }
}
// Define Student class, subclass of Person
public class Student extends Person {
   // Instance variables
   private int numCourses;   // number of courses taken so far, max 30
   private String[] courses; // course codes
```

```
   private int[] grades;     // grade for the corresponding course codes
   private static final int MAX_COURSES = 30; // maximum number of courses

   // Constructor
   public Student(String name, String address) {
      super(name, address);
      numCourses = 0;
      courses = new String[MAX_COURSES];
      grades = new int[MAX_COURSES];
   }

   @Override
   public String toString() {
      return "Student: " + super.toString();
   }

   // Add a course and its grade - No validation in this method
   public void addCourseGrade(String course, int grade) {
      courses[numCourses] = course;
      grades[numCourses] = grade;
      ++numCourses;
   }

   // Print all courses taken and their grade
   public void printGrades() {
      System.out.print(this);
      for (int i = 0; i < numCourses; ++i) {
         System.out.print(" " + courses[i] + ":" + grades[i]);
      }
```

```
      System.out.println();
   }

   // Compute the average grade
   public double getAverageGrade() {
      int sum = 0;
      for (int i = 0; i < numCourses; i++ ) {
         sum += grades[i];
      }
      return (double)sum/numCourses;
   }
}
// Define class Teacher, subclass of Person
public class Teacher extends Person {
   // Instance variables
   private int numCourses;   // number of courses taught currently
   private String[] courses; // course codes
   private static final int MAX_COURSES = 10; // maximum courses

   // Constructor
   public Teacher(String name, String address) {
      super(name, address);
      numCourses = 0;
      courses = new String[MAX_COURSES];
   }

   @Override
   public String toString() {
      return "Teacher: " + super.toString();
```

```
    }

    // Return false if duplicate course to be added
    public boolean addCourse(String course) {
       // Check if the course already in the course list
       for (int i = 0; i < numCourses; i++) {
          if (courses[i].equals(course)) return false;
       }
       courses[numCourses] = course;
       numCourses++;
       return true;
    }

    // Return false if the course does not in the course list
    public boolean removeCourse(String course) {
       // Look for the course index
       int courseIndex = numCourses;
       for (int i = 0; i < numCourses; i++) {
          if (courses[i].equals(course)) {
             courseIndex = i;
             break;
          }
       }
       if (courseIndex == numCourses) { // cannot find the course to be removed
          return false;
       } else {  // remove the course and re-arrange for courses array
          for (int i = courseIndex; i < numCourses-1; i++) {
             courses[i] = courses[i+1];
          }
```

```
            numCourses--;
            return true;
         }
      }
}
// A test driver program for Person and its subclasses
public class Test {
   public static void main(String[] args) {
      // Test Student class
      Student s1 = new Student("Tan Ah Teck", "1 Happy Ave");
      s1.addCourseGrade("IM101", 97);
      s1.addCourseGrade("IM102", 68);
      s1.printGrades();
      System.out.println("Average is " + s1.getAverageGrade());

      // Test Teacher class
      Teacher t1 = new Teacher("Paul Tan", "8 sunset way");
      System.out.println(t1);
      String[] courses = {"IM101", "IM102", "IM101"};
      for (String course: courses) {
         if (t1.addCourse(course)) {
            System.out.println(course + " added.");
         } else {
            System.out.println(course + " cannot be added.");
         }
      }
      for (String course: courses) {
         if (t1.removeCourse(course)) {
            System.out.println(course + " removed.");
```

```
         } else {
            System.out.println(course + " cannot be removed.");
         }
      }
   }
}
```

```
Student: Tan Ah Teck(1 Happy Ave) IM101:97 IM102:68
Average is 82.5
Teacher: Paul Tan(8 sunset way)
IM101 added.
IM102 added.
IM101 cannot be added.
IM101 removed.
IM102 removed.
IM101 cannot be removed.
```