```java
package class9;

import java.util.ArrayList;

public class Driver {

    public static void main(String[] args) {
        // example data member manipulation method
        Student studentOne = new Student("Grace", true);

        // example instance "action" method
        System.out.println(studentOne.toString());

        // example using "getter" method
        if (studentOne.isPostBaccalaureate()) {
            System.out.println(studentOne.toString() + " and I am a postbacc.");
        } // end if

        // example using instanceof conditional test
        if (studentOne instanceof Student) {
            System.out.println(studentOne.toString() + " and I am a student.");
        } // end if

        // == v/s .equals() for reference types
        int a = 77;
        int b = 77;
        int c = 78;
        System.out.println("PRIMITIVE == CHECKS");
        if (a == b) { System.out.println("They're equal!"); }
        if (a == c) { System.out.println("They're equal!"); }

        // must use new operator to avoid compiler optimizations AKA interning
        String nameOne = new String("Jane Smith");
        String nameTwo = new String("Jane Smith");
        String nameThree = new String("Jeff Hill");
        System.out.println("REFERENCE == CHECKS");
        if (nameOne == nameTwo) { System.out.println("They're equal!"); }
        if (nameOne == nameThree) { System.out.println("They're equal!"); }
        System.out.println("REFERENCE .equals() CHECKS");
        if (nameOne.equals(nameTwo)) { System.out.println("They're equal!"); }
        if (nameOne.equals(nameThree)) { System.out.println("They're equal!"); }

        // Professor example
        Professor profOne = new Professor("Jeff","Hill","B123456");
        Professor profTwo = new Professor("Geoffrey","Hill","B123456");
        Professor profThree = new Professor("Jim","Downey","B789012");
        System.out.println(profOne);
        System.out.println(profTwo);
        System.out.println(profThree);
        System.out.println("PROFESSOR == CHECKS");
        if (profOne == profTwo) {System.out.println("They're equal!"); }
        if (profOne == profThree) {System.out.println("They're equal!"); }
        System.out.println("PROFESSOR .equals() CHECKS");
        if (profOne.equals(profTwo)) {System.out.println("One and two are equal!"); }
        if (profOne.equals(profThree)) {System.out.println("One and three are equal!"); }

        // .hashCode() example
```

```
59          System.out.println("PROFESSOR .hashCode() CHECKS");
60          System.out.println(profOne.hashCode());
61          System.out.println(profTwo.hashCode());
62          System.out.println(profThree.hashCode());
63
64          // 1D data structure example
65          System.out.println("1D DATA STRUCTURE EXAMPLE");
66          //ArrayList<String> allNames = new ArrayList<String>(Arrays.asList(new
  String[]{"Jane","Steve","Ada","Jeff"}));
67          ArrayList<String> allNames = new ArrayList<String>();
68          allNames.add("Jane");
69          allNames.add("Steve");
70          allNames.add("Ada");
71          allNames.add("Jeff");
72          for (String eachOne: allNames) { System.out.println(eachOne); }
73
74          // 2D data structure example
75          System.out.println("2D DATA STRUCTURE EXAMPLE");
76          ArrayList<Professor> allProfs = new ArrayList<Professor>();
77          allProfs.add(new Professor("Jeff","Hill","B123456"));
78          allProfs.add(new Professor("Brent","Hill","B654987"));
79          allProfs.add(new Professor("Jim","Downey","B789012"));
80          for (Professor eachOne: allProfs) { System.out.println(eachOne); }
81
82          // .compareTo() example
83          System.out.println("PROFESSOR .compareTo() CHECKS");
84          System.out.println(new Professor("Jeff","Hill","B123456")
85                  .compareTo(new Professor("Brent","Hill","B654987")));
86          System.out.println(new Professor("Jim","Downey","B789012")
87                  .compareTo(new Professor("Jeff","Hill","B123456")));
88          System.out.println(new Professor("Jeff","Hill","B123456")
89                  .compareTo(new Professor("Geoffrey","Hill","B123456")));
90
91          // Collections.sort() example
92          System.out.println("Collections.sort() example");
93          Collections.sort(allProfs);
94          for (Professor eachOne: allProfs) { System.out.println(eachOne); }
95
96          // Collections.reverse() example
97          System.out.println("Collections.reverse() example");
98          Collections.reverse(allProfs);
99          for (Professor eachOne: allProfs) { System.out.println(eachOne); }
100
101         // .contains() example
102         System.out.println("PROFESSOR .contains() CHECKS");
103         System.out.println(allProfs.contains(new Professor("Jeff","Hill","B123456")));
104     } // end main
105
106 } // end Driver
```

```java
package class9;

import java.util.Objects;

public class Professor implements Comparable<Professor> {
    private final String fName;
    private final String lName;
    private final String bearID;

    public Professor(String fName, String lName, String bearID) {
        this.fName = fName;
        this.lName = lName;
        this.bearID = bearID;
    } // end ctor

    public String toString() {
        return "Hi, my name is " + this.fName + " " + this.lName +
                " and my Bear ID is: " + this.bearID;
    } // end toString

    public boolean equals(Object inc) {
        if (inc instanceof Professor) {
            return this.bearID.equals(((Professor)inc).bearID);
        } else {
            return false;
        } // end else
    } // end equals

    public int hashCode() {
        return Objects.hash(this.bearID);
    } // end hashCode

    public int compareTo(Professor inc) {
        return this.bearID.compareTo(inc.bearID);
    } // end compareTo
} // end class Professor
```

Student.java

```java
package class9;

public class Student {
    private final String fName;
    private final boolean postBaccalaureate;

    public Student(String fName, boolean postBaccalaureate) {
        this.fName = fName;
        this.postBaccalaureate = postBaccalaureate;
    } // end ctor

    public String toString() {
        return "Hi, my name is " + this.fName;
    } // end toString

    public boolean isPostBaccalaureate() {
        return this.postBaccalaureate;
    } // end isPostBaccalaureate

} // end Student
```