

# CSCI 145 -- PA 10

## Inheritance

---

Feel free to discuss and help each other out but does not imply that you can give away your code or your answers! You can work with a lab partner for this assignment. **You must always use the required template (JavaClassTemplate.java from Canvas) and output "Author: Your Name(s)" or "Modified by: Your Name(s)" for each program as applicable.**

You can work with a lab partner and each one must submit the same PDF file (include both names in the submission file). Each person must include a brief statement about your contribution to this assignment.

Perform as many exercises from chapter 9 of lab book as possible, but the following lab exercises must be completed. You are not required to turn in written answers to various questions, but it is very helpful in understanding important concepts. You might see those questions on quizzes and exams.

- **Exercise 1** – Exploring Inheritance – remove *static* keyword from *avgBreedWeight()* method
- **Exercise 2** – Sorted Integer List – do not need to include **IntList.java**; do not sort the list every time in the add() method and use a single loop to find the right spot to insert the value.
- **Exercise 3** -- Overriding the equals Method – add the following lines of code to the end of main() method and try it out. If it doesn't compile or run correctly then you probably didn't correctly override the equals method.

```
Object p1 = player1;
Object p2 = player2;
// should work like player1.equals(player2)
System.out.println ("equals: " + p1.equals(p2));
```

### Exercise 4 – Course Information

Define a **Course** base class with methods to set and get the courseNumber and courseTitle. Also define a derived class **OfferedCourse** with methods to set and get instructorName, term, and classTime.

```
public class Course {

    // TODO: Declare private fields - courseNumber, courseTitle

    // TODO: Define default constructor - default to "unknown"
```

```

// TODO: Define mutator methods -
//      setCourseNumber(), setCourseTitle()

// TODO: Define accessor methods -
//      getCourseNumber(), getCourseTitle()

// TODO: Override toString()

}

public class OfferedCourse extends Course {
    // TODO: Declare private fields - instructorName, term, classTime

    // TODO: Define default constructor - default to "unknown"

    // TODO: Define mutator methods -
    //      setInstructorName(), setTerm(), setClassTime()

    // TODO: Define accessor methods -
    //      getInstructorName(), getTerm(), getClassTime()

    // TODO: Override toString()
}

```

Define an application to perform the following:

- Create one Course object and one OfferedCourse object using default constructor
- input data for one Course object and one OfferedCourse object
- output the two objects using System.out.println("Some description:\n" + someObject);

Sample Input/Output:

```

Enter course number: CSCI 145<E>
Enter course name: Java Programming<E>

Enter offered course number: CSCI 145 (20400)<E>
Enter offered course name: Java Programming<E>
Enter offered course instructor: T. Vo<E>
Enter offered course term: Spring 2023<E>
Enter offered course date/time: MW - 1:15-4:15 pm<E>

Course Information:
    Course Number: CSCI 145

```

Course Title: Java Programming

Offered Course Information:

Offered Course Number: CSCI 145 (20400)

Offered Course Name: Java Programming

Offered Course Instructor: T. Vo

Offered Course Term: Spring 2023

Offered Course Date/Time: MW - 1:15-4:15 pm

**Question 1:** Is it a good idea to override instance variables? Explain why or why not.

**Question 2:** Given an existing class X and you would like to reuse it to create a new class, when do you utilize inheritance instead of aggregation?

**Extra Credit:** Set up a new version of **BiasedCoin** like the one from PA8 Exercise 1 but this class inherits from Coin class. Add new data and override appropriate method(s) for this new class. Set up main to perform the same task as PA8 Exercise 1. Try 3 different coins: a fair coin, a coin with 10% head, and a coin with 75% head. You should be able to confirm it is correct with 100 flips of each one.

- a fair coin (50% head) – about 50 heads
- a coin with 10% head – about 10 heads
- a coin with 75% head – about 75 heads

**Fill out and turn in the PA submission file for this assignment (save as PDF format).**