Name: _____ Lab Section: 04

CS161 Fall 2015 Programming Exam

Write your name above and return this paper to your lab instructor when you submit your exam. You must finish and submit your work using Check-In by the end of this lab.

Do your own work. Only look at your own computer screen. Do not browse, email, or use cell phones and other digital technology other than the computers in the lab.

During the exam you may use your Java text book (or ebook version if you let your instructor know), the Java API documentation website and the course website, but no other web pages.

Your grade will be based on:

- Correctness of each method
- Clarity of code and comments. Your code should be easy to read, and if there are non-obvious steps, they should be explained with comments.
- Object-oriented design: Does your code capture the idea of encapsulation
- Testing: Does your main method adequately test all aspects of your code?

You will write four java classes for keeping track of courses you have taken at CSU in different departments. The names of the classes are Transcript, Course, CS, and IndStudy. The classes CS and IndStudy must must extend the class Course to allow an instance of Transcript to use a single ArrayList to hold instances of all courses.

Your classes must implement the following methods.

Transcript: class that is the primary application that you will run.

[**4 points**] **constructor:** Creates an empty ArrayList for holding items of type Course.

[**4 points**] **add:** Add a new course that is either a CS, or IndStudy course.

[**4 points**] **toString:** Returns a string representation of all courses that have been added.

[**3 points**] **main:** A main method that tests each method in this class.

[**3 points**] **Course:** parent class of the CS and IndStudy class.

[**4 points**] **constructor:** Has three arguments, an int for the course number, a String for the semester taken, and an int for the year taken. These values are assigned to protected instance variables.

[**3 points**] **getNumber:** Returns course number.

[**3 points**] **setNumber:** Changes the value of number.

[**3 points**] **getSemester:** Returns the semester taken.

[**3 points**] **setSemester:** Changes the value of semester taken.

[**4 points**] **equals:** Returns true if two objects being compared have the same number. This must correctly override the default equals method.

[**3 points**] **toString:** Returns a string like `Course(333,"spring",2015)` where 333 is the course number, and "spring" and 2015 are the semester and year taken.

[**2 points**] **main:** A main method that tests each method in this class.

[**3 points**] **CS:** child class of Course

[**4 points**] **constructor:** Has four arguments, an int for the course number, a String for the semester taken, an int for the year taken, and a double for the grade as a percent. Assigns these values to protected instance variables.

[**4 points**] **getGrade:** Returns the grade.

[**4 points**] **setGrade:** Changes the value of the grade.

[**3 points**] **equals:** Returns true if two objects being compared have the same name. Implement this only if the method inherited from Course is not sufficient.

[**3 points**] **toString:** Returns a string like `CS(333,"spring",2015)`

[**2 points**] **main:** A main method that tests each method in this class.

[**3 points**] **IndStudy:** child class of Course

[**4 points**] **constructor:** Has four arguments, an int for the course number, a String for the semester taken, an int for the year taken, and a String for the grade as an "S" or "U". Assigns these values to protected instance variables.

[**4 points**] **getGrade:** Returns the grade.

[**4 points**] **setGrade:** Changes the value of the grade.

[**3 points**] **equals:** Returns true if two objects being compared have the same name. Implement this only if the method inherited from Course is not sufficient.

[**3 points**] **toString:** Returns a string like `IndStudy(475,"fall",2014,"S")`

[**2 points**] **main:** A main method that tests each method in this class.

Remember, you are required to implement a `main` method in every class that tests the methods of that class. After your tests in the main methods of each class work, change the main method in your Transcript class to perform the following steps.

[**4 points**] Compiles.

1. Add these courses to your Transcript in the order given.
   - CS161, taken in the fall of 2015 with a grade of 95.2.
   - CS270, taken in the spring of 2016 with a grade of 86.2.
   - IndStudy222, taken in the summer of 2016 with a grade of "U".
   - CS200, taken in the fall of 2016 with a grade of 81.2. written about in 1640.

[**3 points**] 2. Print the transcript.

[**3 points**] 3. Add 2 to the grade for the 2nd item in the transcript by using get method for ArrayLists and get and set methods for courses.

[**1 points**] 4. Print the transcript again.

Here are some reminders about how to use ArrayLists.

- Remember to `import java.utils.ArrayList;`
- Constructing an instance: `ArrayList<Type> list = new ArrayList<Type>();`
- Add an object `o` of type Type doing `list.add(o)`
- Get element at index `i` by doing `list.get(i)`

When you are done, combine your files into a **jar file** named `transcript.jar`

```
jar cvf transcript.jar *.java
```

and check-in your jar file using the Checkin page at CS161 web site.

And you must turn in this exam form with your name on it to your instructor before you leave recitation.