Your Name: _____

Lab 4a:

Show your instructor when you have finished, and ask for help if you are stuck.

Part 1:

Create an interface called **Employable**.

It has the following methods:

public DressCode getDressCode();

public boolean isPaidSalary();

public boolean postSecondaryEducationRequired();

public String getWorkVerb();

Create an abstract class **Employee** which defines one abstract method:
public double getOverTimePayRate(). It also has a **String name** instance variable.
Do not forget the mutator and accessor for **name** and the Constructor**.**
The abstract class **Employee** also implements **Employable**.

Create the following classes which descend from class **Employee**; their respective return values for the Employable-interface methods are listed here.

HockeyPlayer (JERSEY, true, false, "play");	OVERTIME_PAY_RATE: 0.0
Professor (FANCY, true, true, "teach");	OVERTIME_PAY_RATE: 2.0
Parent(ANYTHING, false, false, "care");	OVERTIME_PAY_RATE: -2.0
GasStationAttendant(UNIFORM, false, false, "pump");	OVERTIME_PAY_RATE: 1.5

DressCode should be an Enumeration with 4 values defined:

- JERSEY("jersey")
- FANCY("fancy")
- ANYTHING("anything")
- UNIFORM("uniform")

Define the following instance variables for these classes:

HockeyPlayer:	int numberOfGoals
Professor:	String teachingMajor
Parent:	int numberOfHoursSpentPerWeekWithKids
GasStationAttendant:	double numberOfDollarsStolenPerDay

Create a class called **Employees** which has an ArrayList of 20 Employee references. The constructor adds five of each type of Employee above, in this order:

HockeyPlayer	Wayne Gretzky	scored 894 goals
HockeyPlayer	Who Ever	scored 0 goals
HockeyPlayer	Brent Gretzky	scored 1 goal
HockeyPlayer	Pavel Bure	scored 437 goals
HockeyPlayer	Jason Bourne	scored 0 goals
Professor	Albert Einstein	teaches Physics
Professor	Alan Turing	teaches Computer Systems
Professor	Richard Feynman	teaches Physics
Professor	Tim Berners-Lee	teaches Computer Systems
Professor	Kurt Godel	teaches Logic
Parent	Tiger Woods	spends 1 hour/week with kids
Parent	Super Mom	spends 168 hours/week with kids

Parent	Lazy Larry	spends 20 hours/week with kids
Parent	Ex Hausted	spends 168 hours/week with kids
Parent	Super Dad	spends 167 hours/week with kids

GasStationAttendant	Joe Smith	steals 10 dollars a day
GasStationAttendant	Tony Baloney	steals 100 dollars a day
GasStationAttendant	Benjamin Franklin	steals 100 dollars a day
GasStationAttendant	Mary Fairy	steals 101 dollars a day
GasStationAttendant	Bee See	steals 1 dollar a day

Demonstrate your completed project to your instructor.

Checked by: _____

Lab 4b:

Part 2:

The Employee subclasses must override the Object class's equals method, and also must implement the Comparable interface.

To implement the Comparable interface, the following relationships must be defined:

HockeyPlayers with the most goals are considered "highest".

Professors who teach Computer Science are considered "highest"; others are equal.

Parents who spend the most hours/week with their kids are considered "highest".

GasStationAttendants who steal the most per week are considered "highest".

NOTE: Create a **separate** ArrayList for each child class!

Use Collections.sort(yourArrayList) to sort each of your four Employees collections. Print them out before and after sorting. The Collections.sort() method automatically uses the compareTo() method you defined.

To override the equals method, each of the four Employee subclasses must implement the following method:

```
@Override
public boolean equals(Object that)
```

Which returns the following values:

- true, if this == that
- false, if that == null
- false, if that is not an instanceof the same class (.getClass() != .getClass())
- true for HockeyPlayers if and only if they score the same number of goals
- true for Professors if and only if they teach the same subject
- true for Parents if and only if they parent the same number of hours
- true for GasStationAttendants if and only if they steal the same amount.

The Employees class must contain a method that displays all objects that are equal to one another.
Whenever you override the equals() method you must also override the hashCode() method. Let Eclipse do it for you.

- Demonstrate your completed project to your instructor.
- Checked by: _____