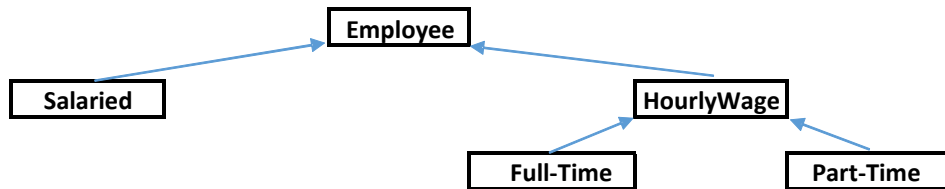Objective: To be able to use inheritance in program design.

Overview: Implement a hierarchy of classes to represent the employee categories:

▪ Salaried Employees
The pay-rate is a weekly salary amount. Salaried employees do not have a time-card and do not earn overtime.

▪ Hourly-Wage Employees
The pay-rate is an hourly wage amount. An hourly-wage employee has a time-card and earns overtime pay. Overtime hours are paid at 1.5 times the hourly pay-rate.

  ▪ Full-Time Employees earn overtime pay for hours over 40 hours in a week.
  ▪ Part-Time Employees are always paid at minimum-wage, and earn overtime pay for hours over 10 hours in any single day.

Specific Requirements
1. Complete the implementation of the TimeCard class.
2. Complete the implementation of the *Employee* class.
   ▪ The constructor must assign a unique number for each new *Employee* instance.
   ▪ *Employee* specifies an **abstract** *grossPay()* method for subclass implementation.
   ▪ In *toString()*, the pay-rate should be *qualified* based on the employee's category, e.g. *$900.00**/Week*** or *$12.50**/Hour***.
3. Complete the implementation of the *SalariedEmployee* class.
   ▪ The *constructor* must enforce a minimum salary constraint. A salaried employee must be paid at least the minimum hourly wage for a full work-week.
4. Complete the implementation of the *HourlyWageEmployee* class.
   ▪ Provide a parameter-less constructor that defaults *payRate* to minimum wage.
   ▪ The parameterized constructor must enforce a minimum wage constraint.
   ▪ Must specify an **abstract** *overtimeHours()* method for subclass implementation.
5. For all classes
   ▪ Provide all required constructors, accessors and mutators.
   ▪ Override *toString()* and *equals()* whenever required.
   ▪ Shared method implementations must always be done in the parent class.
   ▪ Instance variable shadowing is not allowed
6. **Make sure you understand the client/tester**. It is complete except that you must *relax the /* */ comment-delimiters after implementing your subclass constructors*.

Submitting your Assignment
Include a Program Identification Paragraph at the start of each source file. Upload your source files in SCIS Moodle by the due date. **Moodle will not allow late submissions**.