**Online Lab 7**

**Object Oriented Programming**

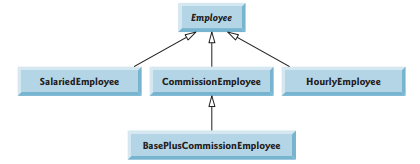
**Learning Objectives**

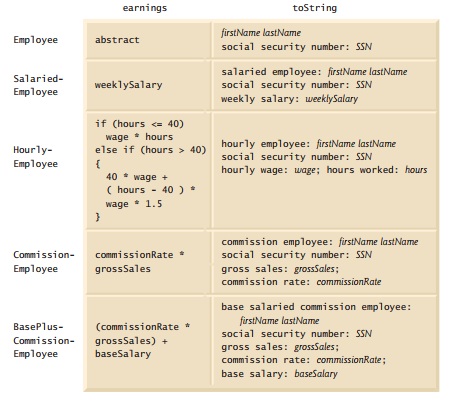
1. The students will be able to

* Implement the inheritance relationship between classes having an Is-A relationship
* Override and reuse base class behavior in the derived class
* Override the common object class methods such as toString.
* Implementing abstract methods and abstract classes
* Handling child classes objects polymorphically
* Downcasting
* Initialization block and static block

|  |
| --- |
| **Lab Walkthrough/Demo** |

**A company pays its employees on a weekly basis. The employees are of four types: Salaried employees are paid a fixed weekly salary regardless of the number of hours worked, hourly employees are paid by the hour and receive overtime pay for all hours worked in excess of 40 hours, commission employees are paid a percentage of their sales and salaried- commission employees receive a base salary plus a percentage of their sales. For the current pay period, the company has decided to reward salaried-commission employees by adding 10% to their base salaries. The company wants to implement a Java application that performs its payroll calculations polymorphically.**

****

****

|  |
| --- |
| **Home Tasks/Lab Assignment** |

* Modify the above payroll system to include private instance variable birthDate in class Employee. Add *get* methods to class Date. Assume that payroll is processed once per month.
* Include an additional Employee subclass PieceWorker that represents an employee whose pay is based on the number of pieces of merchandise produced. Class PieceWorker should contain private instance variables wage (to store the employee’s wage per piece) and pieces (to store the number of pieces produced). Provide a concrete implementation of method earnings in class PieceWorker that calculates the employee’s earnings by multiplying the number of pieces produced by the wage per piece. Create an array of Employee variables to store references to the various employee objects. In a loop, For each Employee, display its String representation and calculate the payroll for each Employee (polymorphically), and add a $100.00 bonus to the person’s payroll amount if the current month is the one in which the Employee’s birthday occurs or an employee object is of type BasePlusCommissionEmployee