Liskov substitution principle means
If S is a subtype of T, then objects of type T maybe replaced with objects of type S.
We first check parent class for Worker and Management, which is AbstractEmployee.
Because Worker and management are both abstract class, we can know that all the methods from AbsrtactEmployee would hold true for Worker and management.
Then we check the subclasses for Management, which are Dispatcher, Manager and Supervisor.
Dispatcher, Manager and Supervisor all inherits the methods from Management and don't override any methods except constructor. And this holds true for Liskov substitution principle.
Last, we check the subclasses for Worker, which are Cashier, WarehouseWorker and Driver.
WarehouseWorker and Driver override the method amountDue(int),

In such case, the implementation of `amountDue(int overtimeHours)` in `Worker` will be:

```
@Override
public amountDue(int overtimeHours){
   if (overtimeHours > allowedOvertimeHours) throw new IllegalOvertimeException(ILLEGAL_OVERTIME_MSG);

   return getRate() * overtimeHours + amountDue();
}
```

We can know that the precondition for Worker class has no limitations for work hours. However, it turns out to be

an exception if we work more hours in the WarehouseWorker class. Thus the precondition for the Worker and WarehouseWorker are not the same.

In a nutshell, Liskov substitution principle holds false for this solution.