CS6308 – JAVA PROGRAMMING

FIRST LABORATORY TEST

DATE: 23.03.23 TIME: 1 HOUR

1. Create a class called **Time** with members: **hrs**(int),**min**(int) and **sec**(int). the methods are:
   - **i. Time()**
   - **ii. Time(int,int,int)**
   - iii. **addHrs(int)-** adds the argument passed to the hrs
   - iv. **addMin(int)** – adds the argument passed to the minutes which increments the hrs if minutes exceed 60
   - v. **addSec(int)** – adds the argument passed to the seconds which increments the minutes and hours appropriately
   - **vi. int getHrs()** – returns the hrs
   - **vii. int getMin()**– returns the min
   - **viii. int getSec()**– returns the sec
   - ix. **toString()** – to display the details of the object Time

   Write a driver class to create objects of **Time** and its functionalties. (10)

2. Write an abstract class called **LendingLibrary** whose protected fields are: **title**(String), **acquisitionTime**(Time), **rentalFee**(double), **dueTime**(Time), bool **checkOut**. The methods are:
   - i. **LendingLibrary(String,double)** that initialises the **title** and **rentalFee** with the arguments passed, **acquisitionTime** and **dueTime** with default values, **checkOut** as false.
   - ii. **void issue(Time)** that sets the acquisitionTime and checkout as true
   - iii. **void return(Time)** – sets checkout as false and calculates whether returned before dueTime and calculates the fine, if any and adds to the rentalFee.
   - **iv. abstract void CalculateDueTime()**
   - v. **toString()** that displays the details (4)

   Define class called **DVD** derives from **LendingLibrary** with member **playLength**(int) and overriding method **CalculateDueTime(int)** that sets the dueTime by adding 3 hours and 30 minutes to the **acquisitionTime** and toString() (3)
   Define class called **Magazine** derives from **LendingLibrary** with members: **volumeNo**(int), **issueNo**(int) and overriding method **CalculateDueTime**(int) that sets the **dueTime** by adding 2 days to the **acquisitionTime** and **toString()** (3)
   Implement the following interface **Invoice** by class **LendingLibrary**: (2)

```
public interface Invoice{
    public static final double fine = 5;
    abstract double calculateFine(Time);
}
```

   fine is charged for every extra 30 minutes. **calcualteFine(Time)** which finds whether the Time returned matches with the **dueTime** if equal returns zero or else returns the fineAmount using the **fine**.

   Write a test application to invoke the functionalities of the each class. (1)