COURSE: (SL-2002) SOFTWARE DESIGN AND ARCHITECTURE LAB
LAB TASK # 8

1. Imagine you are working on a project for a company that wants to manage their employees data. Your task is to create a program that can handle different types of employees such as hourly employees and salaried employees. Each employee has a name, and a salary.

   To achieve this task, you need to create a class hierarchy that uses inheritance to represent the different types of employees. You should start by creating a base class called "Employee" that contains common attributes (“Employee name”) and methods (constructor, getters, setters, and calcSalary()) for all types of employees.

   Next, create derived classes for each type of employee (HourlyEmployee and SalariedEmployee), which inherit from the base class. These derived classes should include additional attributes and methods that are specific to each type of employee.

   For example, the HourlyEmployee class might have attributes for its hourly rate and hours worked, while the SalariedEmployee class might have an attribute for its annual salary. Each derived class should also have a constructor that initializes its specific attributes and calls the base class constructor.

   Finally, you should override the method "calcSalary()" for each derived class. The "calcSalary()" method should calculate the salary for each type of employee based on its specific attributes.

   For Hourly Employee salary calculation:
   Salary= no of hours worked * hourly rate

   For Salaried Employee salary calculation:
   Salary= annual salary/12

   To test your program, you should create one instance of each type of employee and call the "calcSalary()" method to ensure that your program is working correctly.

   Note: Input the employee data from user in main and pass them to the constructor. The salary will be calculated based on the info user has entered.

2. Create a Java program that models a simple role-playing game using polymorphism. The game consists of three types of characters: warriors, mages, and thieves. Each character has a name and a unique set of abilities. Warriors are strong and attack with swords, mages are intelligent and attack with fireballs, and thieves are agile and attack with daggers.

   The Character class is the superclass of the three character types and has a name instance variable and an empty attack() method. The subclasses Warrior, Mage, and Thief inherit from the Character class and each one implements its own unique version of the attack() method, as well as an additional instance variable (strength for Warrior, intelligence for Mage, and agility for Thief).

   Your task is to create a Main class that creates instances of each character type and calls their attack() method using polymorphism. Each character's unique ability should also be displayed along with their name. For example, "John attacks with strength 10" for a warrior named John. Use type casting to access each character's unique ability.
