Core Java

Task Sheet

General Instructions

You will have the afternoon to complete these tasks.

Create a new project for all of the following tasks. As a suggestion, you can call your main project “tasks” but you can call it whatever suits you.

Each task will go into its own separate package and will be run individually. Each task will require its own `main` method. This would mean the following structure with each `App.java` file having its own `main`.

```
|- com.sparta
        |
        |- day1
        |     |- debug
        |     |     |- DebugApp.java
        |     |
        |     |- calculator
        |           |- CalculatorApp.java
        |
        |- day2
```

For all tasks you should be able to explain and justify the solution that you arrive at.

Core Java

Tasks

Animal House

Package: com.sparta.day6.animal

Class: AnimalApp (with other classes)

Create an abstract class `Animal` with the following members:
Fields:

- name (type: `String`) - for example "Fluffy" or "Fido"
- birthdate (type: `LocalDate`)

Methods:

- a constructor with parameters name, year, month, day
- a `getName` method that returns the name
- a `setName` method that has a parameter `newName` and changes the animals name to `newName`
- a `getAge` method that returns an integer - the age of the animal in years
- an override of the `toString` method
- an abstract method `speak` that returns a `String`

Create at least 2 concrete subclasses that extends the abstract class. Each subclass should:

- Provide a constructor that takes the appropriate parameters (including those for the base class)
- For at least one of the subclasses, has an additional field specific to this subclass – for example, `favouriteGame` for a dog.
- Implement the abstract method (`speak`) to return an appropriate `String` – for example "woof, woof" for a dog
- Overrides the `toString` method to include all appropriate information

Next, create an `ArrayList` of `Animal` type and add several objects of each subclass to it. Iterate through the list and print the result of each
object's `speak()` and `toString()` methods.

Core Java

Simple Account

Package: com.sparta.day6.account

Class: AccountApp, BankAccount

Class: BankAccount

- Private fields:
  - String accountNumber
  - String accountHolder
  - double balance
- Constructors:
  - Default constructor
  - Parameterized constructor to set all fields
- Methods:
  - void deposit(double amount)
    → Increase balance, reject negative or zero deposits.
  - void withdraw(double amount)
    → Decrease balance only if sufficient funds exist.
  - void displayInfo()
    → Print account number, holder name, and balance.

Class: AccountApp

- Create 2–3 bank accounts.
- Perform several deposits/withdrawals.
- Display final details.

Bonus challenge:

- Add interestRate (e.g., 2.5%).
- Add a method applyInterest() that increases the balance.