WIX1002 Fundamentals of Programming

Lab Report 1

Question1 & 2: Role System of the Arcane Guild [Instructions for Question 1 and 2]

Problem Statement:

🏰 Background

In the magical kingdom, there exists an ancient and powerful organization — the Arcane Guild.

Every member of the guild has a name, age, race, and some magical power.

Your task is to design a simple program to represent these guild members.

Question1: Define a Class: Role

Package: com.guild.roles.Role

Instance Variables (Encapsulated):

- **private String name; // Name of the role**
- **private Integer age; // Age (can be null)**
- **private String race; // Race (e.g., "Human", "Elf", "Orc")**
- **private Double mana; // Mana (magic energy)**

Requirements:

1. **Define a no-argument constructor that sets default values.**
2. **Define a parameterized constructor to assign all values.**
3. **Provide getter and setter methods for all variables.**
4. **Define a method public void performAction() that prints the role's basic action, e.g. System.out.println(name + " is performing a magical action.");**

Question 2: Test Class: RoleTest

Package: com.guild.roles.RoleTest

Requirements:

1. **Create several Role objects using both constructors.**
2. **Use setter methods to modify some attributes.**
3. **Call performAction() for each object.**
4. **Print out all role details (name, age, race, mana).**

Question 3 & 4: MagicShield Class for Evan

Problem Statement:

Package: com.maplestory.MagicShield

🌟 Background Story

In the world of MapleStory, Evan is preparing for battle against the Black Mage. He decides to craft his own MagicShield.

The shield's size, thickness, and elemental type will affect its defense power and mana consumption.

Question 3: Class Definition Requirements

Instance Variables (Encapsulated)

- **private double radius – Shield radius (in meters, null indicates the shield is inactive)**
- **private double thickness – Shield thickness (in centimeters, null indicates not set)**
- **private String elementType – Elemental attribute of the shield (e.g., "Fire", "Ice", "Light", "Dark")**

Static Variables

- **private static final double DEFAULT_RADIUS = 1.0 – Default radius**
- **private static final double DEFAULT_THICKNESS = 5.0 – Default thickness**
- **private static int shieldCount = 0 – Counter for shields created**

Constructors

1️⃣ **No-argument constructor**

- **Initialize radius and thickness to default values**
- **Set elementType to "Neutral"**
- **Increment shieldCount**

2️⃣ **Parameterized constructor**

- **Accepts radius, thickness, and elementType**
- **Initialize the corresponding fields and increment shieldCount**

Encapsulation Methods

- **getRadius() / setRadius(double radius)**
  - **Throw IllegalArgumentException("Invalid radius") if radius is negative**
- **getThickness() / setThickness(double thickness)**
  - **Throw IllegalArgumentException("Invalid thickness") if thickness is negative**
- **getElementType() / setElementType(String type)**
  - **If type is null or empty, set it to "Neutral"**

Instance Methods

- **double calculateDefensePower()**
  - **Formula:**

    $$\text{Defense Power} = (\text{radius} \times \text{thickness}) \times \text{element coefficient}$$

  - **Element Coefficient Table:**

Element	Coefficient
Fire	1.1
Ice	1.2
Light	1.3

Dark	**1.4**
Neutral	**1.0**

- **double calculateManaCost()**
    - **Formula: manaCost = radius × 10 + thickness × 2**

Class Methods (Static)

- **static int getShieldCount() – Returns total shields created**
- **static boolean isValidSize(double size) – Checks if a size is valid (non-negative and not null)**
- **static double calculateDefensePower(double radius, double thickness, String elementType) – Computes defense power directly for any input**
- **static double calculateManaCost(double radius, double thickness) – Computes mana cost directly for any input**

toString Method

Output format example:

```
[MagicShield Info]
Element Type: Fire
Radius: 2.5 m
Thickness: 7.0 cm
Defense Power: 19.25
Mana Cost: 31.0
```

Question 4: Test Class (com.maplestory.MagicShieldTest) Requirements

1. **Create multiple MagicShield objects using different constructors**
2. **Test setters with valid and invalid values (verify exceptions)**
3. **Calculate and print defense power and mana cost for different shields**
4. **Use static methods to calculate shield properties for arbitrary parameter combinations**
5. **Output the total number of shields created (verify the static counter)**
6. **Test edge cases: null, 0, negative values**

Question 5& 6: Student System of the Arcane Academy

Problem Statement:

🏫 **Background Story**

In the magical kingdom, there is an ancient Arcane Academy, which trains future mages and alchemists.
Every student has a name, age, and mana level.
The academy needs to track the total number of students and determine whether two students are exactly the same (e.g., students with identical names and ages are considered the same).

Question 5: Base Class: Person (Student Base Class)

Package: com.magical.people.Person

Instance Variables (Encapsulated)

- **private String name – Student name**
- **private int age – Student age (null indicates unknown age)**

Static Variables

- **private static final int DEFAULT_AGE = 18 – Default age, used for no-arg constructor**
- **private static int personCount = 0 – Tracks total number of students in the academy**

Constructors

1️⃣ **No-argument constructor**

- **Initialize name = "Unknown Student"**
- **Initialize age = DEFAULT_AGE**
- **Increment personCount**

2️⃣ **Parameterized constructor**

- **Accept name and age parameters and assign them to fields**
- **Increment personCount**

Encapsulation Methods

- **getName() / setName(String name)**
- **getAge() / setAge(int age)**

- Validate that age is not negative; if negative, throw IllegalArgumentException

Class Methods

- **static int getPersonCount() – Returns the total number of students created**
- **public boolean compareTo(Person other)**
  - **Returns true if both name and age are identical between the current student and other; otherwise, returns false**

Question 6: Test Class: PersonTest

Package: com.magical.people.PersonTest

Requirements

1️⃣ **Create multiple Person objects using both no-arg and parameterized constructors**

2️⃣ **Call setAge() with valid and invalid ages and verify that exceptions are thrown for invalid input**

3️⃣ **Call compareTo() to compare different students and output whether they are exactly the same**

4️⃣ **Call getPersonCount() to verify that the total student count in the academy is correct**

INSTRUCTIONS :

1. **Complete all questions in your designated project group.**
2. **All members must contribute to writing the codes. (i.e. 1 question = 1 person, and share the workload if there's an additional question relative to the actual number of members in your team (i.e. 5)). Ensure that all members must understand and explain codes from any of the questions.**
3. **During viva, all students in each team will be randomly asked to describe, answer and edit any of the answers provided. Marks will be given to your ability to present the answers.**

Lab Report

Prepare a report to solve the above problems. The report should contain all the sections as below for each question:

No	Section	Description
1	Problem	Description on the problem
2	Solution	Explanation on how to solve the above problems
3	Sample Input & Output	A few sets of input and output (snapshot)
4	Source Code	Java Source Code

Requirements

1. Group Assignment (4-5 students per group)

2. Cover page that includes all student matric number and full name.

3. Font: Times New Roman 12, Line Spacing: 1 1⁄2 Spacing

4. Submit to Spectrum according to your OCC.