

Lab worksheet 3: Introduction

Instructions

1. Create a Java project in IntelliJ within your folder and name it using your student number in the following formats: "**OOP_CS_2022_XXX**" or "**OOP_ET_2022_XXX**". (Eg: OOP_CS_2022_001, OOP_ET_2022_007)
2. Create distinct packages for each lab worksheet, naming them in the following format "**LW_XX**." (Eg: LW_01)
3. Create distinct classes for each question, naming them "**QX**." (Eg: Q1)
4. Upload your project files to your GitHub repository.

Questions

1. Define a new class named **Temperature**. Save the temperature internally in Celsius using a private double variable **celsius**. The class has two Constructors one is a No-Arg Constructor, and another one is Parameterized Constructor which accepts a double type argument celsius. The class also has two getter/accessor methods, **toFahrenheit** and **toCelsius** that return the temperature in the specified unit, and two setter/mutator methods **setFahrenheit** and **setCelsius** that assign the temperature in the specified unit. Using this class, write a Java program that accepts temperature in Celsius as an input using the Scanner class and outputs the temperature in equivalent Fahrenheit.

$$\text{celsius} = (\text{fahrenheit} - 32) * 5 / 9$$
$$\text{fahrenheit} = \text{celsius} * 9 / 5 + 32$$

2. Using the **setFahrenheit** method of **Temperature** class from the above Exercise, write a Java program that accepts temperature in Fahrenheit as an input using the **Scanner** class and outputs the temperature in equivalent Celsius.
3. Modify the **Bicycle** class provided below, to assign an owner object instead of just the owner's name and telephone number. The new **Owner** class will have two data members **ownerName** and **phoneNo**.

File: Bicycle.java

```
class Bicycle {
    // Data Member
    private String ownerName;
    private String phoneNo;

    //Constructor: Initializes the data member
    public Bicycle() {
        ownerName = "Unknown";
    }
}
```

Lab worksheet 3: Introduction

```
}

public Bicycle(String name,String num) {
    ownerName = name;
    phoneNo = num;
}

//Returns the name of this bicycle's owner
public String getOwnerName() {
    return ownerName;
}

//Assigns the name of this bicycle's owner
public void setOwnerName(String name) {
    ownerName = name;
}

public String getPhoneNo() {
    return phoneNo;
}

//Assigns the name of this bicycle's owner
public void setPhoneNo(String num) {
    phoneNo = num;
}
}
```

4. Please write a Java program for a university course registration system. The program should consist of the following classes: **Course**, **Lecturer**, **Student**, and **Main**.

In the **Course** class, include two String variables: **courseName** and **courseCode**. Additionally, include a **lecturer** object along with their corresponding getter and setter methods.

The **Lecturer** class should have two String variables named **lecturerName**, **courseTeaching** along with their getter and setter methods.

The **Student** class should include three String variables: **studentName**, **degreeName**, and **courseFollowing**. It should also have corresponding getter and setter methods.

Lastly, implement a **Main** class to serve as the entry point of the program. Create **course**, **lecturer**, and **student** objects, and assign them all the necessary properties.

Lab worksheet 3: Introduction

5. Imagine you are building a library system using Java to manage books and other library items. The system will involve the creation of classes for **BorrowableItems**, **Book**, **Person**, and **Student**. These classes will help you showcase OOP principles.

Create an abstract class called **BorrowableItems** with an abstract method **displayInfo()**.

Create a class named **Book** as a subclass of the **BorrowableItems** class with the following private attributes:

- **title** (String)
- **author** (String)
- **ISBN** (String)
- **available** (boolean)

Provide public getter and setter methods for these attributes.

Include a constructor to initialize the attributes.

Implement the **displayInfo()** method that displays information about the book.

Create a **Person** class with the following private attribute:

- **name** (String).

Create a constructor to set the **name**.

Create a **Student** class that inherits from the **Person** class.

Create a **Library** class with the following private attribute:

- **BorrowableItemsList** (Array or another suitable data structure to store **BorrowableItems** objects)

Implement methods to:

- Add library items to the collection.
- Check out library items (mark them as available or unavailable).
- List all available library items in the library.
- Display information about the library, including the number of available items.

Create instances of the **Book** class representing library books. Use the following book information:

- Title: "Seena Maragatham", Author: "Sujatha Thilaka", ISBN: "9789553023975"
- Title: "Nuramakadya Bauthika Nuladanayakshanamaala", Author: "Seynamasasekaka", ISBN: "9789553548721"
- Title: "Island of a Thousand Mirrors", Author: "Nayomi Munaweera", ISBN: "9781616953623"

Create an instance of the **Library** class to manage the library system.

Lab worksheet 3: Introduction

Add the created books to the library's collection using the **addLibraryItem** method of the **Library** class. For each book, call **addLibraryItem** to include them in the library's inventory.

Display information about the library, including the total number of items, by calling the **displayLibraryInfo** method.

List all available library items in the library using the **listAvailableItems** method of the **Library** class.

Demonstrate the process of checking out a library item by calling the **checkoutItem** method. For example, check out the book with the title "Island of a Thousand Mirrors" by setting its availability to false.

List available items in the library again using the **listAvailableItems** method. Observe that the book you checked out is now marked as unavailable.