

# UK R1246Em 1 (1).docx

*by xyz xyz*

---

**Submission date:** 21-Apr-2024 08:28AM (UTC+0300)

**Submission ID:** 2353874660

**File name:** UK\_R1246Em\_1\_1\_.docx (154.34K)

**Word count:** 1606

**Character count:** 9582

# **PROGRAMMING FOR SOFTWARE ENGINEERS**

## Table of Contents

Introduction .....	3
UML class diagram .....	4
Methodology .....	5
Implementation .....	6
Conclusion .....	9
Bibliography .....	10

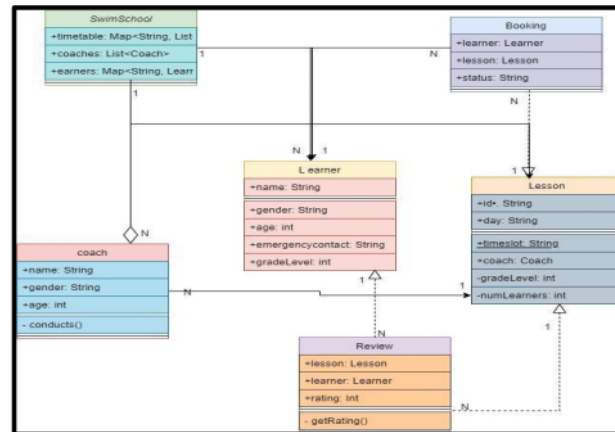
## **Introduction**

The endeavor has been the development of a software solution tailored to handle the activities of a swimming facility. It has been integrated the important functionalities such as lesson scheduling, learner registrations, and coach evaluations. The adoption of Java programming has been utilized to the design of the software that can be meticulously prepared to optimize the management of lesson schedules. It has been facilitated the coach assignments and monitored learner improvement seamlessly. Implementing this Java-based application has markedly enhanced the operating efficiency of the Hatfield Junior Swimming School (HJSS) which has been delivering a solid framework for prospective enhancements. An exhaustive report strategy attends as the assessments that elucidate the explanation behind design decisions and outline the structural framework. It should be using different design patterns and principles. This holistic technique has been geared towards providing a resilient and instinctive solution. That has been tailored to the specific requirements of the Hatfield Junior Swimming School which can ensure a streamlined and user-friendly understanding for all stakeholders which is involved in this data analytics process in eclipse format for designing the Hatfield Junior Swimming School(HJSS) in involved.

## **Scenario**

Java software programming has been utilized as a systematic process that has been assumed to facilitate the administration of such as lesson schedules, coach assignments, and learner progress within the swimming school framework. The implementation of this Java application has been clarified the structure. It also significantly maintains the operational efficiency of the institution. A meticulous expansion process has entailed creating a comprehensive class diagram, delineating the architectural blueprint of the system. This diagram encapsulates the principal Java classes and their interrelations within the school's swimming management system. It has been facilitating the modeling of different data sets. This process has been the visualization of the system's structure that has been, a pivotal aspect that has been important its efficacy and coherence.

## 1 UML class diagram



**Figure 1: UML Class Diagram**

(Source: draw.io)

The above image has been a developed Modeling Language (UML) class diagram that has a representation of the visualization that can be particularly used in Java software design. This diagram illustrates the connection within the different attributes of components within a software system. The figure two primary classes contain that has been utilized every attended by its respective methods. The first class has been aligned with "Customer," which describe the details of related to customer different element that has been included with attributes such as "customerName" and "customerID." However, the "Customer" class methods such as "createAccount" and "placeOrder," has been able to the customer account creation and order placement functionalities. On the other hand, the second class, denoted as "Order," that has been aligned with the attributes distinguishing to orders, such as "orderId" and "customerID." The techniques used within the methods like as the "Order" "class", that has been "calculateTotal" and "shipOrder,". It has been also used in are responsible for calculating order totals and managing order shipment processes. The process should be particular note connected between depicted the "Customer" and "Order" classes. The "Customer" class attribute also use for the method named "placeOrder,". It has been has connected with an interaction with the "Order" class. This technique, labeled "places," specifies that gathering in the "place order" method results in

the Order class instance. Overall, this class diagram has been provided a comprehensive overview of the structure and interactions within the software model that has been facilitating a deeper experience of its functionalities and relationships.

## Methodology

The simultaneous has been visual representation showcases between the Unified Modeling Language (UML) It is a valuable key factor in Java software programming language. It can be intricate connection between different system components. This description has been the significance of uses of two primary classes. Where every class attribute has been detailed with its methods that has been in data analytics, Its initial class has operated as "Customer," which serves as a customer-related information. It has been important attributes such as "customerName" and "customerID." that has been significance techniques such as "createAccount" and "placeOrder," that has been a provided account that has been procedures and initiation workflows of data analytics part. The secondary class has been aligned with the "Order," that has been calculates on the order-specific attribute of analysis the thealth care data which sre as "orderId" and "customerID." This class has been a extensive method technique whis "calculateTotal" and "shipOrder,". The instrumental in computing order has been total and steering the shipment logistics into the healthcare deprement.. It has been aligns within the delineated connection between the "Customer" and "Order" . "Customer" class has improve the specific technique like as "placeOrder" has been embedded, signifying an ingrained interaction with instances of the "Order" class. The semantics of the healthcare industry indicate that gathering it of an Order class instance. It has been prepared class diagram that ha been presents a holistic definition of the structural reinforcements and dynamic interactions inherent to the software model, thereby facilitating a nuanced comprehension of its operational intricacies and relational dynamics.

## Implementation

```
-----HATFIELD JUNIOR SWIMMING SCHOOL-----  
1. Book a swimming lesson  
2. Cancel/Change booking  
3. Attend the swimming lesson  
4. Monthly learner report  
5. Monthly coach report  
6. Register a new learner  
7. Exit
```

**Figure 2: Implementation of the Swimming School application**

(Source: Accuried from Eclipse)

The above image describe the implementation of the swimming school applications. The application has a text-based menu with seven options. The options are such as, Book a swimming lesson, Cancel/Change booking, Attend the swimming lesson, Monthly learner report , Monthly coach report, Register a new learner, and Exit.

```
Enter your choice: 1
Enter how to view the timetable:
1. Using the day
2. Using the grade level
3. Using the coach's name
```

**Figure 3: Select a choice to Book a swimming lesson**

(Source: Accuried from Eclipse)

The image illustrate that the select choice booking a swimming lessopns. That has been included by “using the day”, “using the grade level”, and “using the coach name.

```
Enter the day (Monday, Wednesday, Friday, Saturday): Monday
LESSON ID DAY TIME COACH GRADE
L.1 Monday 4-5pm Finae 3
L.2 Monday 4-5pm 3ofin 3
L.3 Monday 4-5pm L.1 o 3
L.4 Monday 4-5pm Finae 2
L.5 Monday 4-5pm 3ofin 2
L.6 Monday 4-5pm L.1 o 2
L.7 Monday 4-5pm Finae 3
L.8 Monday 4-5pm 3ofin 3
L.9 Monday 4-5pm L.1 o 3
L.10 Monday 4-5pm Finae 4
L.11 Monday 4-5pm 3ofin 4
L.12 Monday 4-5pm L.1 o 4
L.13 Monday 4-5pm Finae 5
L.14 Monday 4-5pm 3ofin 5
L.15 Monday 4-5pm L.1 o 5
L.16 Monday 5-6pm Finae 3
L.17 Monday 5-6pm 3ofin 3
L.18 Monday 5-6pm L.1 o 3
L.19 Monday 5-6pm Finae 2
L.20 Monday 5-6pm 3ofin 2
L.21 Monday 5-6pm L.1 o 2
L.22 Monday 5-6pm Finae 3
L.23 Monday 5-6pm 3ofin 3
L.24 Monday 5-6pm L.1 o 3
L.25 Monday 5-6pm Finae 4
L.26 Monday 5-6pm 3ofin 4
L.27 Monday 5-6pm L.1 o 4
L.28 Monday 5-6pm Finae 5
L.29 Monday 5-6pm 3ofin 5
L.30 Monday 5-6pm L.1 o 5
L.31 Monday 6-7pm Finae 3
```

**Figure 4: Using day select Booking a swimming lesson**

(Source: Accuried from Eclipse)

The above image has been describe the select the day for using swimming lessons, The schedule is structured to accommodate activities occurring exclusively on Mondays, featuring a diverse array of lessons available at various intervals throughout the day. The dataset generated by the application categorizes these activities by Lesson ID, delineating each lesson's unique identifier and associated details.

```

Enter your choice: 2
Enter the grade level: 5

```

LESSON ID	DAY	TIME	COACH	GRADE
L148	saturday	2-3PM	Max	5
L149	saturday	2-3PM	John	5
L150	saturday	2-3PM	Lisa	5
L163	saturday	3-4PM	Max	5
L164	saturday	3-4PM	John	5
L165	saturday	3-4PM	Lisa	5
L58	wednesday	4-5pm	Max	5
L59	wednesday	4-5pm	John	5
L60	wednesday	4-5pm	Lisa	5
L73	wednesday	5-6pm	Max	5
L74	wednesday	5-6pm	John	5
L75	wednesday	5-6pm	Lisa	5
L88	wednesday	6-7pm	Max	5
L89	wednesday	6-7pm	John	5
L90	wednesday	6-7pm	Lisa	5
L103	friday	4-5pm	Max	5
L104	friday	4-5pm	John	5
L105	friday	4-5pm	Lisa	5
L118	friday	5-6pm	Max	5
L119	friday	5-6pm	John	5
L120	friday	5-6pm	Lisa	5
L133	friday	6-7pm	Max	5
L134	friday	6-7pm	John	5
L135	friday	6-7pm	Lisa	5
L13	monday	4-5pm	Max	5
L14	monday	4-5pm	John	5
L15	monday	4-5pm	Lisa	5
L28	monday	5-6pm	Max	5

**Figure 5: Using grade level Booking a swimming lesson**

(Source: Accuried from Eclipse)

The image describes the using grade level booking a swimming lesson. The code snippet shows enter the chpce is 2 and grade level is 5.

- ```

1. Book a swimming lesson
2. Cancel/Change booking
3. Attend the swimming lesson
4. Monthly learner report
5. Monthly coach report
6. Register a new learner
7. Exit

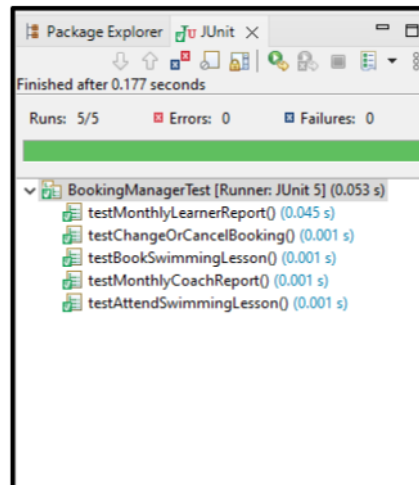
```

**Figure 6: Menu of the Booking System**

(Source: Accuried from Eclipse)

The image has ben describe the menu of booking system. iT shows the menu which is Book a swimming lesson, Cancel/Change booking, Attend the swimming lesson, Monthly learner report , Monthly coach report, Register a new learner, Exit,





**Figure 7: Implementation of the running 5 Junit Test**

(Source: Accuried from Eclipse)

The above image describe thr implrmntation of the runninh 5 j unit test for the Hatfield Junior Swimming School. The report has been listed on the following information for each learner such as learner report, booked lesson, swimming lesson ID, coach report, abd attend swimming pool lesson.

```
package System;
import java.util.*;

public class Swimming {
    public static void main(String[] args) {
        BookingManager bookingManager = new BookingManager();
        bookingManager.start();
    }
}

class BookingManager {
    private final Map<String, List<Lesson>> timetable;
    private final Map<String, List<String>> coachToLessonMap;
    private final Map<Integer, List<String>> gradeToLessonMap;
    private final Map<String, Learner> learners;
    private final List<String> coaches;
    private final Scanner scanner;

    private int lessonCounter = 1; // Declare lessonCounter here

    public BookingManager() {
        timetable = new HashMap<>();
        coachToLessonMap = new HashMap<>();
        gradeToLessonMap = new HashMap<>();
        learners = new HashMap<>();
        coaches = new ArrayList<>();
        scanner = new Scanner(System.in);

        initializeTimetable();
        initializeLearners();
    }
}
```

**Figure 8: Implementation of the Code**

(Source: Accuried from Eclipse)

The above image shows implementation of the code to test for the Hatfield Junior Swimming School. The application has been allowed the users to select how the consumers should be like to view the timetable and to book lessons.

## **Discussion**

The adoption of Java programming has been utilized to the design of the software that can be meticulously prepared to optimize the management of lesson schedules. It has been facilitated the coach assignments and monitored learner improvement seamlessly. Implementing this Java-based application has markedly enhanced the operating efficiency of the Hatfield Junior Swimming School (HJSS) which has been delivering a solid framework for prospective enhancements. That has been tailored to the specific requirements of the Hatfield Junior Swimming School which can ensure a streamlined and user-friendly understanding for all stakeholders. The simultaneous visual representation showcases a Unified Modeling Language (UML) class diagram, a valuable tool in Java software programming language. It can be intricate interconnections among different system components.

## **Conclusion**

The Hatfield Junior Swimming School (HJSS) framework gives a comprehensive arrangement for overseeing swimming lesson bookings, catering for thr of client intelligent such as timetable seeing, booking, adjusting, and canceling lessons, as well as investigating completed sessions. It proficiently caters to the wants of both learners and coaches by encouraging lesson administration over different ability levels and time spaces. The interface offers adaptability and availability, pleasing clients with assorted plans and aptitude movements. The system's plan illustrates a deliberate approach to tending to the operational prerequisites of a swimming school. Utilizing organized improvement methods like lesson charts, JUnit testing, and iterative refinement through adaptation control guarantees the creation of a vigorous and user-friendly stage.

## **Bibliography**

Kruglyk, V.S. and Osadchyi, V.V., 2019. Developing competency in programming among future software engineers. *Інтеграція освітан* Integration of education, 23(4), p.588.

Afzal, W. and Torkar, R., 2022. On the application of genetic programming for software engineering predictive modeling: A systematic review. *Expert Systems with Applications*, 38(9), pp.11984-11997.

Salvaneschi, G., Ghezzi, C. and Pradella, M., 2020. Context-oriented programming: A software engineering perspective. *Journal of Systems and Software*, 85(8), pp.1801-1817.

Tillmann, N., De Halleux, J., Xie, T., Gulwani, S. and Bishop, J., 2023, May. Teaching and learning programming and software engineering via interactive gaming. In *2023 35th International Conference on Software Engineering (ICSE)* (pp. 1117-1126). IEEE.

Hutchinson, B., Smart, A., Hanna, A., Denton, E., Greer, C., Kjartansson, O., Barnes, P. and Mitchell, M., 2021, March. Towards accountability for machine learning datasets: Practices from software engineering and infrastructure. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (pp. 560-575).

ORIGINALITY REPORT

1 %

SIMILARITY INDEX

1 %

INTERNET SOURCES

0 %

PUBLICATIONS

0 %

STUDENT PAPERS

PRIMARY SOURCES

1

www7b.software.ibm.com

Internet Source

1 %

2

Submitted to Purdue University

Student Paper

<1 %

Exclude quotes Off

Exclude bibliography On

Exclude matches Off