Timetable Scheduling System

1. Introduction

Team Details

Name	Roll No	Email
Akshat Batra	IMT2023025	Akshat.Batra@iiitb.ac.in
Dhruv Ramesh Joshi	IMT2023032	dhruvramesh.joshi@iiitb.ac.in
Ankith Kini	IMT2023075	Ankith.Kini@iiitb.ac.in
Arnav Oruganty	IMT2023078	Arnav.Oruganty@iiitb.ac.in
Vaishak Prasad Bhat	IMT2023085	Vaishakprasad.bhat@iiitb.ac.in
Harshit Krishna R	IMT2023613	Harshit.Krishna@iiitb.ac.in

Project Overview

The Timetable Scheduling System is designed to provide an automated solution for managing and organising academic timetables through a command-line interface (CLI). We take file inputs using File I/O and export our output to a CSV file. This system addresses challenges related to manual scheduling, such as conflicting time slots, classroom allocation, and faculty availability. It ensures no scheduling conflicts, such as double-booking classrooms or faculty, occur.

GitHub Repository:

https://github.com/vaishak-iiitb/class-schedule

High-Level Functionality:

- Automated timetable generation.
- Conflict resolution for room and faculty availability.
- Providing batch-wise timetable.
- Scheduling for both academic classes and exams

Scope

Current Capabilities:

- **Make Course Timetable for All Batches in the College:** Develop a fully automated timetable for each batch, avoiding faculty and room allocation conflicts. Implement a

timetable to accommodate all mandatory courses, maintaining consistency across all the batches.

- **Make Timetable for Exams:** Create a separate timetable for exams, ensuring room availability and preventing clashes with regular classes. This schedule will be crafted to suit each batch, allowing students and faculty to focus on exams without impacting the regular curriculum.

Future Capabilities and Additions: (Optional)

- Make a Timetable for Individual Professors based on their availability.
- Optimising the scheduling algorithm
- Develop it such that this system can be used to make other schedules as well like the schedule for the sporting events in our college.
- Handle cases where one classroom cannot accommodate the entire batch (as opposed to the assumption taken here)

---

2. Objectives

- Automate the process of creating class timetables to minimise manual errors.
- Eliminate scheduling conflicts by ensuring faculty and room availability checks.
- Streamline the creation of exam timetables for different batches to reduce manual errors.

---

3. System Overview

Technical Specifications:

- Java for core class definitions and implementations. The algorithm for scheduling has been written in C++. We have used C++ (for class timetable scheduling) and Java (for exam scheduling) to handle input data and make the timetable algorithm. We have used JNI to interact between the C++ and Java components of our project.

Input/Output Requirements:

Input:

- We are giving file inputs - courses, dates and rooms for Exam Scheduling. And courses (batch-wise), professors and rooms for Lecture TimeTable Scheduling.

Output:

- A batch-wise exam schedule for all the batches in list format is displayed via the CLI. With the export to CSV option, we get the timetable as a snapshot of a table similar to the one given by our college.
- Similarly, we get a batch-wise timetable for lectures as a table using the export to CSV option.

---

4. Functional Requirements

Detailed Features:

- Create academic timetables for all batches. (We have assumed all rooms to have the same capacity and that a batch can entirely fit in any room. We have also assumed each batch to have the same number of students.)
- Generate timetables for exams. (We have taken the same assumptions as mentioned above)

Use Cases:

- **Admins use case:** Generate and display both the class timetables and exam schedules for all batches in an organized table format. Export the finalized timetables as ready-to-deploy CSV files for seamless integration and deployment.

---

5. Development Setup

Instructions

Software Requirements:

- C++ compiler (e.g., GCC)
- Java Development Kit (JDK)
- Java Native Interface (JNI)
- IDEs: Visual Studio Code (VSC)

---

6. Workflow

Backend:

- C++ will manage scheduling algorithms and ensure conflict resolution.

- Java will handle the input file reading and pass the required data to Java for processing. (for Exam Scheduling)
- The input and processing takes place in C++ itself for Lecture TimeTable Scheduling.

Output:

- Once we run the program, we will get the output as the timetable for all the batches (both exams and timetable) and simultaneously it will be exported to a CSV file which will be properly formatted and ready to deploy wherever required.

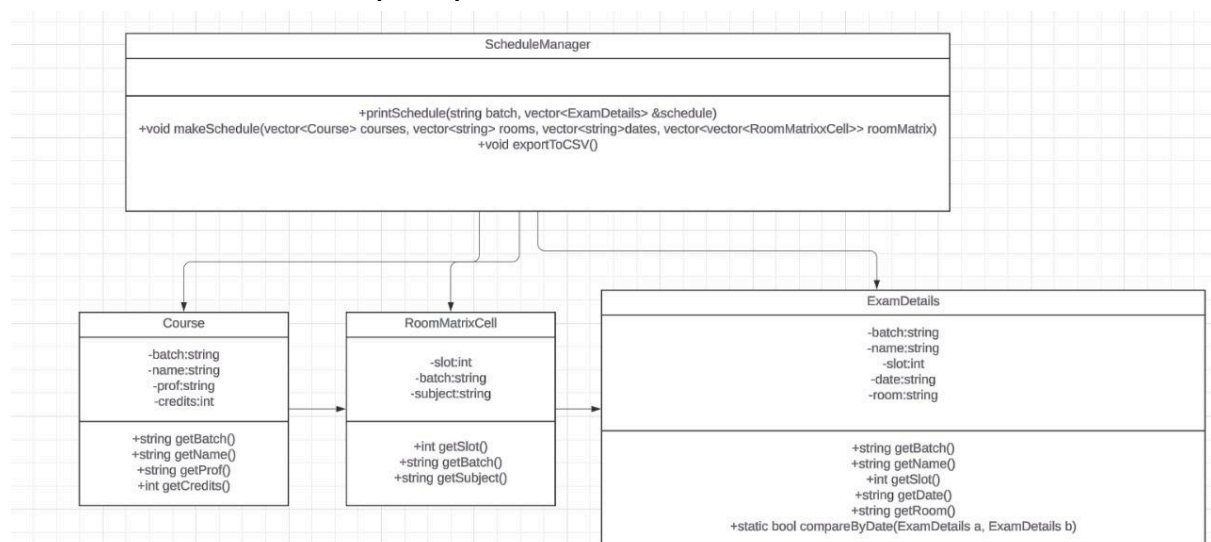# 7. Testing & Logging

Testing Strategy:

- Unit tests for individual Java and C++ classes and methods.
- Integration tests to verify communication between C++ input handling and Java processing.
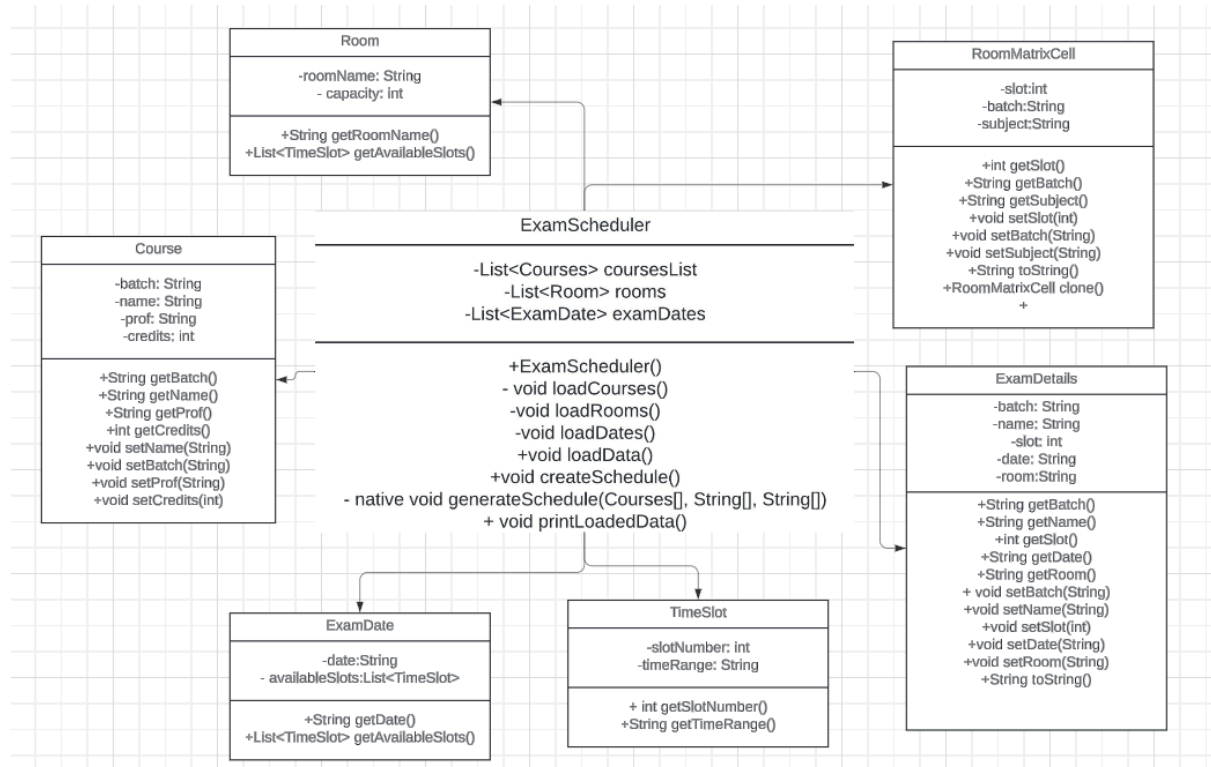
Logging Mechanisms:

- Implement logging in Java to track scheduling operations and errors within the CLI system.
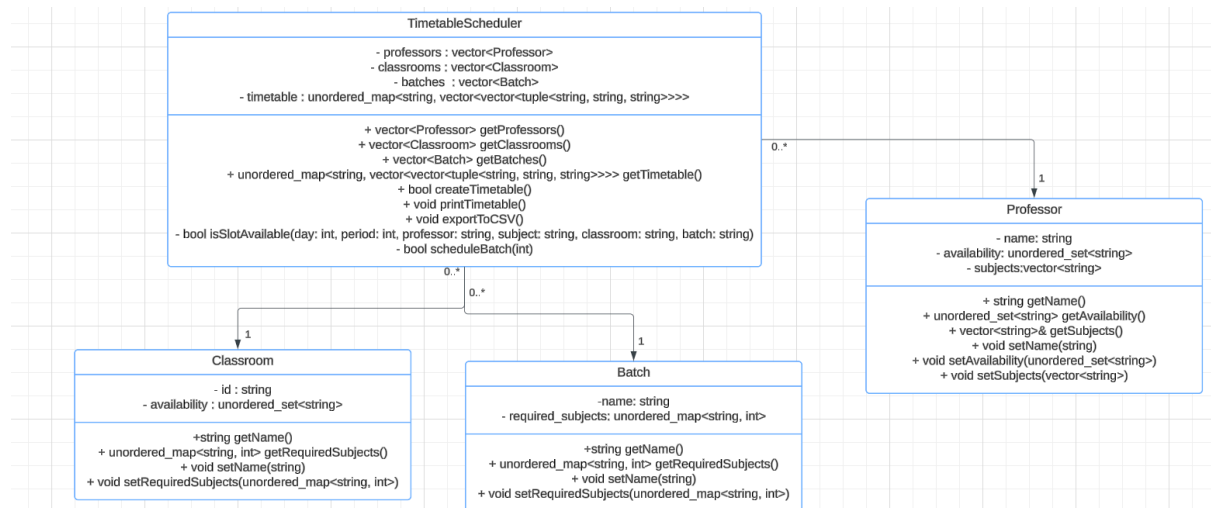
---

8. UML Diagram

Exam Scheduler UML (CPP)

Exam Scheduler UML(Java)

Room

-roomName: String
- capacity: int

+String getRoomName()
+List<TimeSlot> getAvailableSlots()

RoomMatrixCell

-slot:int
-batch:String
-subject:String

+int getSlot()
+String getBatch()
+String getSubject()
+void setSlot(int)
+void setBatch(String)
+void setSubject(String)
+String toString()
+RoomMatrixCell clone()
+

ExamScheduler

-List<Courses> coursesList
-List<Room> rooms
-List<ExamDate> examDates

+ExamScheduler()
- void loadCourses()
-void loadRooms()
-void loadDates()
+void loadData()
+void createSchedule()
- native void generateSchedule(Courses[], String[], String[])
+ void printLoadedData()

Course

-batch: String
-name: String
-prof: String
-credits: int

+String getBatch()
+String getName()
+String getProf()
+int getCredits()
+void setName(String)
+void setBatch(String)
+void setProf(String)
+void setCredits(int)

ExamDetails

-batch: String
-name: String
-slot: int
-date: String
-room:String

+String getBatch()
+String getName()
+int getSlot()
+String getDate()
+String getRoom()
+ void setBatch(String)
+void setName(String)
+void setSlot(int)
+void setDate(String)
+void setRoom(String)
+String toString()

ExamDate

-date:String
- availableSlots:List<TimeSlot>

+String getDate()
+List<TimeSlot> getAvailableSlots()

TimeSlot

-slotNumber: int
-timeRange: String

+ int getSlotNumber()
+String getTimeRange()

Timetable Scheduler UML

TimetableScheduler

- professors : vector<Professor>
- classrooms : vector<Classroom>
- batches : vector<Batch>
- timetable : unordered_map<string, vector<vector<tuple<string, string, string>>>>

+ vector<Professor> getProfessors()
+ vector<Classroom> getClassrooms()
+ vector<Batch> getBatches()
+ unordered_map<string, vector<vector<tuple<string, string, string>>>> getTimetable()
+ bool createTimetable()
+ void printTimetable()
+ void exportToCSV()
- bool isSlotAvailable(day: int, period: int, professor: string, subject: string, classroom: string, batch: string)
- bool scheduleBatch(int)

Professor

- name: string
- availability: unordered_set<string>
- subjects:vector<string>

+ string getName()
+ unordered_set<string> getAvailability()
+ vector<string>& getSubjects()
+ void setName(string)
+ void setAvailability(unordered_set<string>)
+ void setSubjects(vector<string>)

Classroom

- id : string
- availability : unordered_set<string>

+string getName()
+ unordered_map<string, int> getRequiredSubjects()
+ void setName(string)
+ void setRequiredSubjects(unordered_map<string, int>)

Batch

-name: string
- required_subjects: unordered_map<string, int>

+string getName()
+ unordered_map<string, int> getRequiredSubjects()
+ void setName(string)
+ void setRequiredSubjects(unordered_map<string, int>)

0..*

1

0..*

0..*

1

1

9. Input and Output

- ## Timetable Scheduler

Input:

courses_1.txt

```
IMT2024 7
"Calculus" "Manisha Kulkarni" 4
"Modern Economics" "V Sridhar" 2
"Thermodynamics" "B Ashok" 2
"Molecular Biology" "Priyanka Sharma" 4
"Statistics" "Priyanka Das" 2
"Quantum Mechanics" "Shiva M" 2
"Biochemistry" "Kurian Polachan" 4
IMT2023 5a
"Optics" "B Ashok" 4
"Inorganic Chemistry" "Kurian Polachan" 2
"Differential Equations" "Priyanka Das" 4
"Ancient Civilizations" "Amrita Mishra" 2
"Genetics" "Priyanka Sharma" 4
IMT2022 4
"Nuclear Physics" "Shiva M" 4
"Medieval History" "Amrita Mishra" 2
"Linear Algebra" "Manisha Kulkarni" 4
"Renaissance History" "V Sridhar" 2
IMT2021 5
"Astrophysics" "Ashish Choudhary" 4
"Artificial Intelligence" "R Chandrashekar" 2
"Philosophy" "V Muralidhara" 4
"Machine Learning" "G Srinivasaraghavan" 2
"Political Science" "Amit Prakash" 4
IMT2020 6
"Computer Science" "R Chandrashekar" 4
"Organic Chemistry" "Ashish Choudhary" 2
"Evolutionary Biology" "V Muralidhara" 4
"Cognitive Psychology" "Amit Prakash" 2
"Game Theory" "G Srinivasaraghavan" 4
"Cryptography" "Nanditha Rao" 2
```

professors_1.txt

```
"Manisha Kulkarni" "Calculus,Linear Algebra" "(0,0)(0,1)(0,2)(0,3)(1,1)(1,2)(1,3)(2,0)(2,1)(2,3)(3,0)(3,2)(3,3)(4,0)(4,1)(4,2)"
"B Ashok" "Thermodynamics,Optics" "(0,0)(0,1)(0,2)(0,3)(1,0)(1,1)(1,2)(1,3)(2,0)(2,1)(2,2)(2,3)(3,0)(3,1)(3,2)(3,3)(4,0)(4,1)(4,2)(4,3)",
"Kurian Polachan" "Inorganic Chemistry,Biochemistry" "(0,0)(0,1)(0,2)(0,3)(1,0)(1,1)(1,2)(1,3)(2,0)(2,1)(2,2)(2,3)(3,0)(3,1)(3,2)(3,3)(4,0)(4,1)(4,2)(4,3)",
"Priyanka Sharma" "Molecular Biology,Genetics" "(0,0)(0,1)(0,2)(0,3)(1,0)(1,1)(1,2)(1,3)(2,0)(2,1)(2,2)(2,3)(3,0)(3,1)(3,2)(3,3)(4,0)(4,1)(4,2)(4,3)",
"Amrita Mishra" "Ancient Civilizations,Medieval History" "(0,0)(0,1)(0,2)(0,3)(1,0)(1,1)(1,2)(1,3)(2,0)(2,1)(2,2)(2,3)(3,0)(3,1)(3,2)(3,3)(4,0)(4,1)(4,2)(4,3)",
"Priyanka Das" "Differential Equations,Statistics" "(0,0)(0,1)(0,2)(0,3)(1,0)(1,1)(1,2)(1,3)(2,0)(2,1)(2,2)(2,3)(3,0)(3,1)(3,2)(3,3)(4,0)(4,1)(4,2)(4,3)",
"Shiva M" "Quantum Mechanics,Nuclear Physics" "(0,0)(0,1)(0,2)(0,3)(1,0)(1,1)(1,2)(1,3)(2,0)(2,1)(2,2)(2,3)(3,0)(3,1)(3,2)(3,3)(4,0)(4,1)(4,2)(4,3)",
"V Sridhar" "Modern Economics,Renaissance History" "(0,0)(0,1)(0,2)(0,3)(1,0)(1,1)(1,2)(1,3)(2,0)(2,1)(2,2)(2,3)(3,0)(3,1)(3,2)(3,3)(4,0)(4,1)(4,2)(4,3)",
"Ashish Choudhary" "Astrophysics,Organic Chemistry" "(0,0)(0,1)(0,2)(0,3)(1,0)(1,1)(1,2)(1,3)(2,0)(2,1)(2,2)(2,3)(3,0)(3,1)(3,2)(3,3)(4,0)(4,1)(4,2)(4,3)",
"V Muralidhara" "Philosophy,Evolutionary Biology" "(0,0)(0,1)(0,2)(0,3)(1,0)(1,1)(1,2)(1,3)(2,0)(2,1)(2,2)(2,3)(3,0)(3,1)(3,2)(3,3)(4,0)(4,1)(4,2)(4,3)",
"Amit Prakash" "Political Science,Cognitive Psychology" "(0,0)(0,1)(0,2)(0,3)(1,0)(1,1)(1,2)(1,3)(2,0)(2,1)(2,2)(2,3)(3,0)(3,1)(3,2)(3,3)(4,0)(4,1)(4,2)(4,3)",
"R Chandrashekar" "Computer Science,Artificial Intelligence" "(0,0)(0,1)(0,2)(0,3)(1,0)(1,1)(1,2)(1,3)(2,0)(2,1)(2,2)(2,3)(3,0)(3,1)(3,2)(3,3)(4,0)(4,1)(4,2)(4,3)",
"G Srinivasaraghavan" "Machine Learning,Game Theory" "(0,0)(0,1)(0,2)(0,3)(1,0)(1,1)(1,2)(1,3)(2,0)(2,1)(2,2)(2,3)(3,0)(3,1)(3,2)(3,3)(4,0)(4,1)(4,2)(4,3)",
"Nanditha Rao" "Cryptography" "(0,0)(0,1)(0,2)(0,3)(1,0)(1,1)(1,2)(1,3)(2,0)(2,1)(2,2)(2,3)(3,0)(3,1)(3,2)(3,3)(4,0)(4,1)(4,2)(4,3)"s
```

```
"R103" "(0,0)(0,1)(0,2)(0,3)(1,0)(1,1)(1,2)(1,3)(2,0)(2,1)(2,2)(2,3)(3,0)(3,1)(3,2)(3,3)(4,0)(4,1)(4,2)(4,3)"
"R203" "(0,0)(0,1)(0,2)(0,3)(1,0)(1,1)(1,2)(1,3)(2,0)(2,1)(2,2)(2,3)(3,0)(3,1)(3,2)(3,3)(4,0)(4,1)(4,2)(4,3)"
"A106" "(0,0)(0,1)(0,2)(0,3)(1,0)(1,1)(1,2)(1,3)(2,0)(2,1)(2,2)(2,3)(3,0)(3,1)(3,2)(3,3)(4,0)(4,1)(4,2)(4,3)"
"A204" "(0,0)(0,1)(0,2)(0,3)(1,0)(1,1)(1,2)(1,3)(2,0)(2,1)(2,2)(2,3)(3,0)(3,1)(3,2)(3,3)(4,0)(4,1)(4,2)(4,3)"
"A303" "(0,0)(0,1)(0,2)(0,3)(1,0)(1,1)(1,2)(1,3)(2,0)(2,1)(2,2)(2,3)(3,0)(3,1)(3,2)(3,3)(4,0)(4,1)(4,2)(4,3)"
"A305" "(0,0)(0,1)(0,2)(0,3)(1,0)(1,1)(1,2)(1,3)(2,0)(2,1)(2,2)(2,3)(3,0)(3,1)(3,2)(3,3)(4,0)(4,1)(4,2)(4,3)"
```

Output:

Timetable for Batch: IMT2020

Time\Day	Monday	Tuesday	Wednesday	Thursday	Friday
9:30 - 11:00	Free	Free	Computer Science by R Chandrashekar in A204	Game Theory by G Srinivasaraghavan in A106	Free
11:15 - 12:45	Free	Free	Computer Science by R Chandrashekar in R203	Evolutionary Biology by V Muralidhara in R103	Free
1:30 - 3:00	Cognitive Psychology by Amit Prakash in A303	Free	Organic Chemistry by Ashish Choudhary in R103	Free	Evolutionary Biology by V Muralidhara in R203
3:15 - 4:45	Free	Free	Game Theory by G Srinivasaraghavan in A204	Cryptography by Nanditha Rao in A106	Free

Timetable for Batch: IMT2021

Time\Day	Monday	Tuesday	Wednesday	Thursday	Friday
9:30 - 11:00	Political Science by Amit Prakash in R203	Free	Free	Free	Free
11:15 - 12:45	Free	Free	Free	Free	Free
1:30 - 3:00	Free	Machine Learning by G Srinivasaraghavan in A204	Artificial Intelligence by R Chandrashekar in A204	Free	Political Science by Amit Prakash in A204
3:15 - 4:45	Astrophysics by Ashish Choudhary in A303	Philosophy by V Muralidhara in A204	Philosophy by V Muralidhara in A106	Astrophysics by Ashish Choudhary in R103	Free

Timetable for Batch: IMT2022

Time\Day	Monday	Tuesday	Wednesday	Thursday	Friday
9:30 - 11:00	Free	Free	Free	Free	Free
11:15 - 12:45	Nuclear Physics by Shiva M in A106	Renaissance History by V Sridhar in A204	Linear Algebra by Manisha Kulkarni in R103	Free	Free
1:30 - 3:00	Nuclear Physics by Shiva M in R103	Free	Free	Free	Free
3:15 - 4:45	Linear Algebra by Manisha Kulkarni in R103	Free	Medieval History by Amrita Mishra in R103	Free	Free

Timetable for Batch: IMT2023

Time\Day	Monday	Tuesday	Wednesday	Thursday	Friday
9:30 - 11:00	Free	Free	Optics by B Ashok in R103	Free	Genetics by Priyanka Sharma in A106
11:15 - 12:45	Free	Differential Equations by Priyanka Das in A106	Free	Ancient Civilizations by Amrita Mishra in R203	Optics by B Ashok in A204
1:30 - 3:00	Free	Free	Inorganic Chemistry by Kurian Polachan in R203	Free	Free
3:15 - 4:45	Free	Differential Equations by Priyanka Das in A303	Free	Free	Genetics by Priyanka Sharma in A303

Timetable for Batch: IMT2024

Time\Day	Monday	Tuesday	Wednesday	Thursday	Friday
9:30 - 11:00	Free	Biochemistry by Kurian Polachan in R103	Free	Free	Calculus by Manisha Kulkarni in A303
11:15 - 12:45	Thermodynamics by B Ashok in R203	Free	Free	Biochemistry by Kurian Polachan in A305	Molecular Biology by Priyanka Sharma in A106
1:30 - 3:00	Calculus by Manisha Kulkarni in A106	Quantum Mechanics by Shiva M in R203	Molecular Biology by Priyanka Sharma in A303	Free	Free
3:15 - 4:45	Free	Free	Modern Economics by V Sridhar in A305	Statistics by Priyanka Das in A204	Free

● **Exam Scheduler**

Input:

```
IMT2024 7
"Digital Design" "Kurian Polachan" 4
"Economics" "V Shridhar" 4
"English" "Priyanka Sharma" 2
"Programming in C" "Badrinath" 2
"Mathematics-1" "Manisha Kulkarni" 4
"Mathematics-2" "Travis head" 4
"Computational Biology" "Virat Kohli" 2
IMT2023 5
"Discrete Mathematics" "Ashish Choudhary" 4
"Systems and Signals" "Vinod Reddy" 4
"Programming in Java" "Vivek Yadav" 4
"Physics" "Malpakka" 4
"Chemistry" "Walter White" 4
IMT2025 4
"Microprocessors" "Anjali Desai" 4
"Data Structures" "Rajesh Gupta" 4
"Linear Algebra" "Arun Kumar" 4
"Introduction to AI" "Preeti Nair" 2
IMT2026 6
"Computer Networks" "Rahul Mehta" 4
"Operating Systems" "Deepa Sharma" 4
"Engineering Mathematics" "Amitabh Singh" 4
"Digital Signal Processing" "Neha Verma" 4
"Biology for Engineers" "Kavita Joshi" 2
"Chemistry-2" "Teacher 1" 2
IMT2027 5
"Subject 1" "Teacher 2" 4
"Subject 2" "Teacher 3" 4
"Subject 3" "Teacher 4" 4
"Subject 4" "Teacher 5" 2
"Subject 5" "Teacher 6" 2
```

dates.txt

```
01/12/2024
02/12/2024
03/12/2024
04/12/2024
05/12/2024
```

rooms.txt

```
Room1
Room2
Room3
Room4
```

Output:

```
IMT2023:
Name: Physics, Slot: 9:30 am - 12:30 pm, Date: 01/12/2024, Room: Room2
Name: Chemistry, Slot: 9:30 am - 12:30 pm, Date: 02/12/2024, Room: Room2
Name: Discrete Mathematics, Slot: 2:00 pm - 5:00 pm, Date: 03/12/2024, Room: Room1
Name: Systems and Signals, Slot: 2:00 pm - 5:00 pm, Date: 04/12/2024, Room: Room1
Name: Programming in Java, Slot: 2:00 pm - 5:00 pm, Date: 05/12/2024, Room: Room1

IMT2024:
Name: Digital Design, Slot: 9:30 am - 12:30 pm, Date: 01/12/2024, Room: Room1
Name: Mathematics-2, Slot: 2:00 pm - 5:00 pm, Date: 01/12/2024, Room: Room1
Name: Economics, Slot: 9:30 am - 12:30 pm, Date: 02/12/2024, Room: Room1
Name: Computational Biology, Slot: 2:00 pm - 5:00 pm, Date: 02/12/2024, Room: Room1
Name: English, Slot: 9:30 am - 12:30 pm, Date: 03/12/2024, Room: Room1
Name: Programming in C, Slot: 9:30 am - 12:30 pm, Date: 04/12/2024, Room: Room1
Name: Mathematics-1, Slot: 9:30 am - 12:30 pm, Date: 05/12/2024, Room: Room1

IMT2025:
Name: Introduction to AI, Slot: 2:00 pm - 5:00 pm, Date: 01/12/2024, Room: Room2
Name: Microprocessors, Slot: 9:30 am - 12:30 pm, Date: 03/12/2024, Room: Room2
Name: Data Structures, Slot: 9:30 am - 12:30 pm, Date: 04/12/2024, Room: Room2
Name: Linear Algebra, Slot: 9:30 am - 12:30 pm, Date: 05/12/2024, Room: Room2

IMT2026:
Name: Biology for Engineers, Slot: 9:30 am - 12:30 pm, Date: 01/12/2024, Room: Room3
Name: Computer Networks, Slot: 2:00 pm - 5:00 pm, Date: 02/12/2024, Room: Room2
Name: Chemistry-2, Slot: 9:30 am - 12:30 pm, Date: 02/12/2024, Room: Room3
Name: Operating Systems, Slot: 2:00 pm - 5:00 pm, Date: 03/12/2024, Room: Room2
Name: Engineering Mathematics, Slot: 2:00 pm - 5:00 pm, Date: 04/12/2024, Room: Room2
```

```
IMT2027:
Name: Subject 4, Slot: 2:00 pm - 5:00 pm, Date: 01/12/2024, Room: Room3
Name: Subject 5, Slot: 2:00 pm - 5:00 pm, Date: 02/12/2024, Room: Room3
Name: Subject 1, Slot: 9:30 am - 12:30 pm, Date: 03/12/2024, Room: Room3
Name: Subject 2, Slot: 9:30 am - 12:30 pm, Date: 04/12/2024, Room: Room3
Name: Subject 3, Slot: 9:30 am - 12:30 pm, Date: 05/12/2024, Room: Room3

Exam schedules exported to ExamSchedules.csv successfully!
```

10. Conclusion

The Timetable Scheduling System is designed to streamline the scheduling process using a straightforward CLI interface, eliminating the need for manual effort and minimizing errors. Leveraging the object-oriented strengths of Java and C++ and employing C++ for algorithmic efficiency, the system generates optimized timetables swiftly. It ensures effective utilization of faculty and room resources while maintaining accuracy and reliability in scheduling.