Spring 2023
Department of Computer Science
Bahria University, Karachi Campus

Artificial Intelligence

Complex Computing Problem

BS(CS)-6A

Group Members

Name	Enrollment
1. Shahzadi Zainub Mumtaz	02-134202-051
2. Naeem ur Rehman	02-134202-053
3. Ali Gauhar	02-134202-006

Submitted to:

Muhammad Tariq Siddique

a. Algorithms **explained** with their mathematical formulas/flow diagrams of your proposed solution. [CLO1, PLO1, C2]

Timetable Inputs:

1. Number of subjects
2. Number of teachers
3. Maximum lectures a teacher can conduct
4. Priority of subjects and topics
5. Time constraints (e.g., working days of the week)

Constraints:

1. A teacher cannot conduct more than the maximum allowed lectures.
2. All subjects and topics must be covered within the given time constraints.
3. The timetable should be optimized to make the best use of available resources.

Hard constraints:

1. Unique class timings
2. Course.student<=room.seating capacity
3. Multiple classes shouldn’t have the same room at one timeslot
4. Class timing for each teacher is unique

Soft constraints:

1. Classes are allocated according to section requirements
2. All courses are according to their department
3. Even distribution of course in a section per week.

Local Search Algorithms:

Genetic Algorithms:

These algorithms are inspired by the process of natural selection. They work with a population of candidate solutions and evolve them over time by applying genetic operators such as mutation, crossover, and selection. This approach can help explore a larger solution space and avoid local optima.

Mathematical Formula:

The fitness function for the Genetic Algorithm can be defined as:

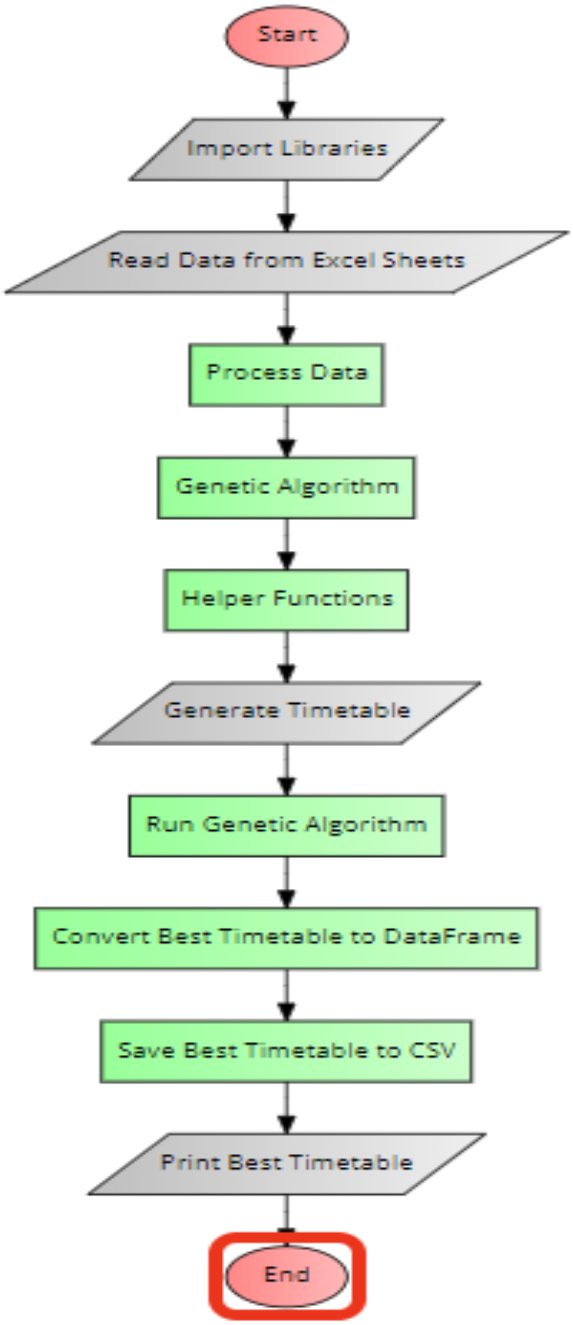
$$\textbf{fitness(s) = w1 * C1(s) + w2 * C2(s) + ... + wn * Cn(s)}$$

where s is a timetable solution, Ci(s) is the ith constraint, and wi is the weight associated with the ith constraint. The algorithm aims to maximize the fitness function.

Algorithm:

1. Initialize a population of timetable solutions.
2. Evaluate the fitness of each solution in the population.
3. Select parents based on their fitness.
4. Apply crossover and mutation operators to generate offspring.
5. Replace the least fit individuals in the population with offspring.
6. Repeat steps 2-5 until a termination condition is met (e.g., a maximum number of generations).

Flow Diagram:


Start
Import Libraries
Read Data from Excel Sheets
Process Data
Genetic Algorithm
Helper Functions
Generate Timetable
Run Genetic Algorithm
Convert Best Timetable to DataFrame
Save Best Timetable to CSV
Print Best Timetable
End

Ant Colony Optimization:

ACO is a metaheuristic optimization algorithm inspired by the foraging behavior of ants. It is commonly used to solve combinatorial optimization problems. Here's an overview of the ACO algorithm:

Initialization:

- Create a set of ants and place them on the problem space.
- Initialize pheromone trails and heuristics for the problem.

Ant Behavior:

- Each ant constructs a solution by probabilistically selecting the next component based on pheromone trails and heuristics.
- Pheromone updates are performed based on the quality of the solutions constructed by the ants.

Pheromone Update:

- The pheromone trails are updated using the pheromone evaporation rate and the pheromone deposit amount based on the quality of the solutions.
- The best solution found so far leaves stronger pheromone trails.

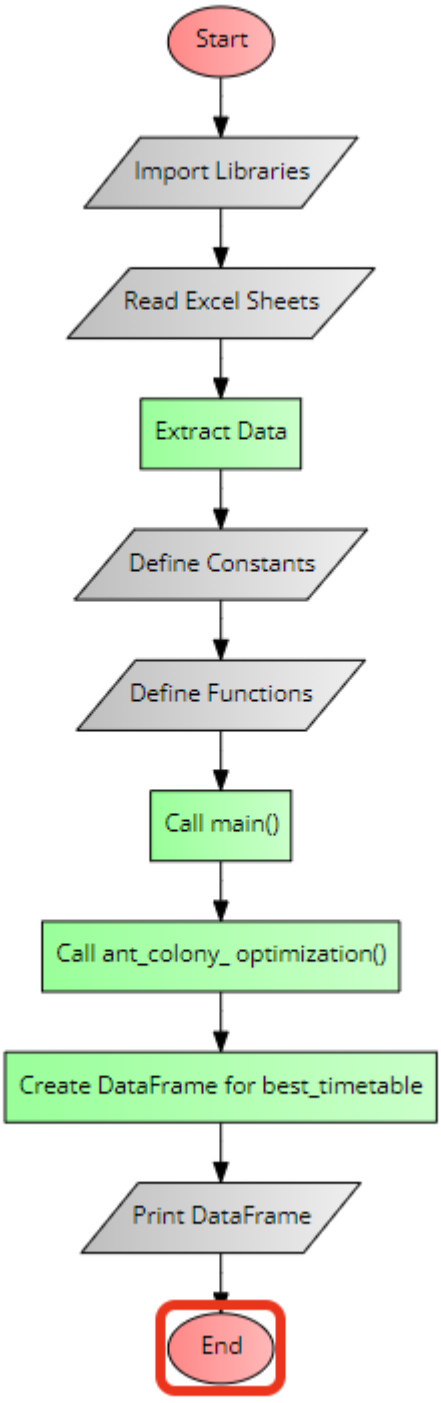
Termination:

- The algorithm terminates when a stopping condition (e.g., a maximum number of iterations) is met.

Mathematical Formulas:

- Probability of choosing a component at position i: $P_i = (t_i^\alpha) * (\eta_i^\beta) / \Sigma((t_k^\alpha) * (\eta_k^\beta))$
- Pheromone update: $\Delta t_i = Q / L$, where Q is a constant, and L is the quality (fitness) of the solution.

Flow Diagram:



Hill-Climbing:

This algorithm starts with an initial solution and iteratively makes small changes to the current solution to find a better one. It continues until no further improvement can be made. The algorithm can be prone to getting stuck in local optima.

Mathematical Formula:

The objective function for the Hill-Climbing algorithm can be defined as:

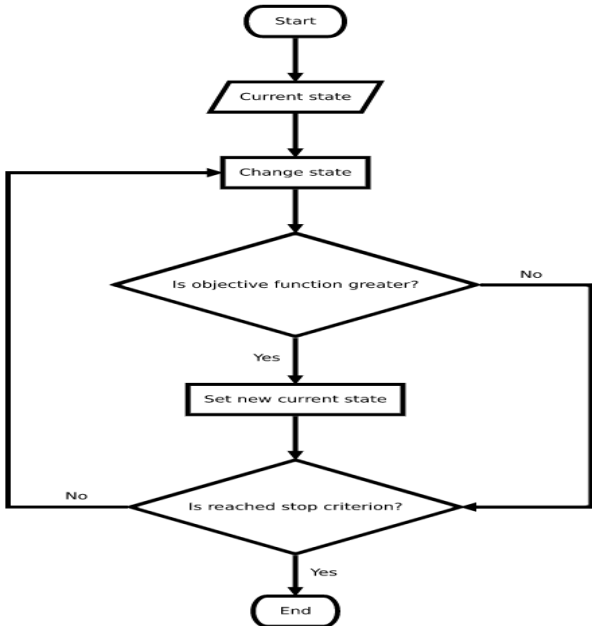
$$f(s) = w1 * C1(s) + w2 * C2(s) + ... + wn * Cn(s)$$

where s is a timetable solution, Ci(s) is the ith constraint, and wi is the weight associated with the ith constraint. The algorithm aims to maximize the objective function f(s).

Algorithm:

1. Start with an initial solution s.
2. Generate a neighboring solution s' by making a small change to s.
3. If f(s') > f(s), set s = s'.
4. Repeat steps 2-3 until no further improvement can be made.

Flow Diagram:



Spring 2023
Department of Computer Science
Bahria University, Karachi Campus

Constraint Satisfaction Problem (CSP) Algorithms:

Minimum Value Remaining (MVR):

This is a variable ordering heuristic used in backtracking search algorithms for CSPs. It selects the variable with the fewest legal values remaining in its domain, which can help reduce the search space and improve efficiency.

Arc Consistency:

This is a technique used to simplify CSPs by enforcing a consistency property called arc consistency. It iteratively removes values from the domains of variables that cannot be part of any solution, thus reducing the search space and making the problem easier to solve.

CSP Backtracking Algorithm:

CSP Backtracking Algorithm is a search algorithm used to solve Constraint Satisfaction Problems (CSPs). It systematically explores the solution space by incrementally assigning values to variables while respecting constraints and backtracks when a conflict or dead end is encountered. It operates recursively, trying out different assignments and undoing them if they lead to inconsistencies. The algorithm explores the search tree using depth-first search, backtracking to previous decision points, when necessary, until a valid solution or failure is found.

Algorithm:

1. Start with an empty assignment (no variables assigned).

2. Select an unassigned variable using a variable ordering heuristic (e.g., Minimum Value Remaining).

3. For each value in the variable's domain, do the following:

   a. Check if the value is consistent with the current assignment (i.e., it doesn't violate any constraints).

   b. If the value is consistent, assign the value to the variable and recursively apply the algorithm to the remaining unassigned variables.

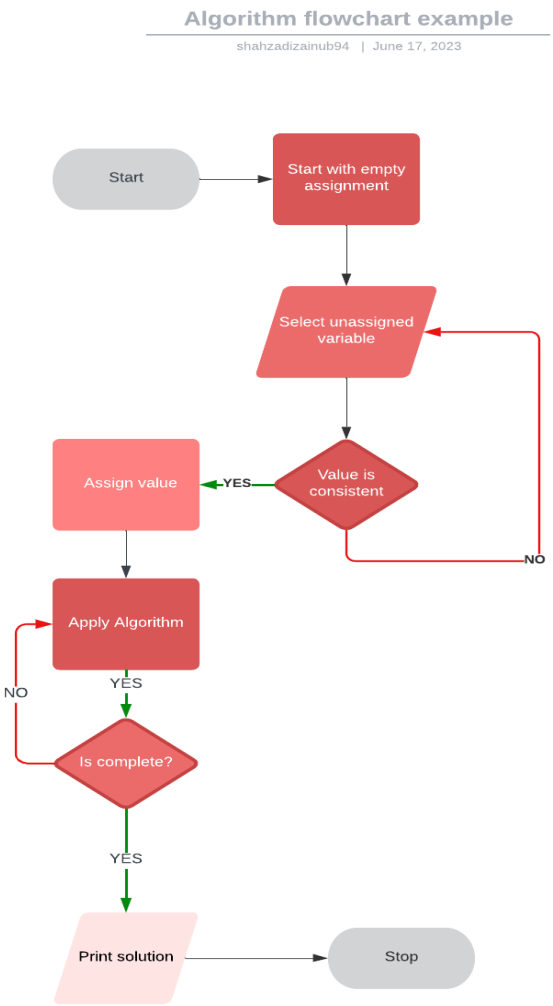
c. If the assignment is complete and consistent, return the assignment as a solution.

d. If the assignment is not complete, backtrack and try the next value in the variable's domain.

4. If no consistent assignment is found for the current variable, backtrack to the previous variable and try the next value in its domain.

5. If all variables have been tried and no solution is found, return failure.

Flowchart:



b. Examine the above scenario and develop Python code for both local search and CSP algorithms. [CLO3, PLO3, C4]

Python Code

Genetic Algorithm Code

```
import pandas as pd
import random
from datetime import datetime, timedelta

def convert_excel_time(excel_time):
    return (datetime(1900, 1, 1) + timedelta(days=excel_time - 2)).time()



filename = "C:/Users/Naeem ur rehman/Desktop/Rooms.xlsx"
teachers_df = pd.read_excel(filename, sheet_name="teacher")
batch_courses_df = pd.read_excel(filename, sheet_name="batch courses")
credit_hours_df = pd.read_excel(filename, sheet_name="credit hours ")
available_rooms_df = pd.read_excel(filename, sheet_name="Rooms")
timeslots_df = pd.read_excel(filename, sheet_name="timeslot")
timeslots_df['Days'] = timeslots_df['Days'].str.lower()
timeslots = timeslots_df.set_index('Days').T.to_dict('list')



teachers = {}
for index, row in teachers_df.iterrows():
    teacher = row["teacher"]
    courses = row["coure"].split(", ")
    teachers[teacher] = courses


batch_courses = {}
for index, row in batch_courses_df.iterrows():
    class_name = row["classes"]
    courses = row[1:].dropna().tolist()
```

```
    batch_courses[class_name] = courses

credit_hours = {}
for index, row in credit_hours_df.iterrows():
    course = row["coures"]
    credit_hour = row["Credit Hours"]
    credit_hours[course] = credit_hour


available_rooms = list(available_rooms_df["Rooms"])


max_teaching_hours = 12


working_days = ["monday", "tuesday", "wednesday", "thursday", "friday"]


population_size = 50
max_generations = 100



def calculate_fitness(timetable):
    fitness_score = 0


    for day in timetable:
        course_count = {}
        for class_details in timetable[day]:
            course = class_details[1]
            if course not in course_count:
                course_count[course] = 0
            course_count[course] += 1

        for count in course_count.values():
            if count > 1:
                fitness_score -= count


    for timeslot in range(3):
        teacher_count = {}
```

```
        for day in timetable:
            for class_details in timetable[day]:
                if class_details[4] == timeslot:
                    teacher = class_details[2]
                    if teacher not in teacher_count:
                        teacher_count[teacher] = 0
                    teacher_count[teacher] += 1

        for count in teacher_count.values():
            if count > 1:
                fitness_score -= count


    for timeslot in range(3):
        room_count = {}
        for day in timetable:
            for class_details in timetable[day]:
                if class_details[4] == timeslot:
                    room = class_details[3]
                    if room not in room_count:
                        room_count[room] = 0
                    room_count[room] += 1

        for count in room_count.values():
            if count > 1:
                fitness_score -= count

    return fitness_score




def genetic_algorithm():

    population = []
    for _ in range(population_size):
        timetable = generate_timetable()
        population.append(timetable)


    generation = 1
```

```
    while generation <= max_generations:

        fitness_scores = []
        for timetable in population:
            fitness_score = calculate_fitness(timetable)
            fitness_scores.append((timetable, fitness_score))


        population = [x[0] for x in sorted(fitness_scores, key=lambda x: x[1],
reverse=True)]


        parents = population[:population_size//2]


        offspring = []
        while len(offspring) < population_size:
            parent1, parent2 = random.sample(parents, 2)
            child = crossover(parent1, parent2)
            child = mutate(child)
            offspring.append(child)


        population = offspring


        generation += 1


    best_timetable = population[0]

    return best_timetable

def generate_timetable():
    timetable = {}
    for day in working_days:
        timetable[day] = []

    for class_name, courses in batch_courses.items():
        for course in courses:
            credit_hour = credit_hours[course]
```

```
            teacher = select_teacher(course, timetable)
            room = select_room()

            if teacher_exceeds_hours(teacher, credit_hour, timetable):
                teacher = select_teacher(course, timetable)

            for _ in range(credit_hour):
                day = random.choice(working_days)
                timeslot = random.randint(0, len(timeslots[day]) - 1)
                time = convert_excel_time(timeslots[day][timeslot])
                timetable[day].append((class_name, course, teacher, room, time))

    return timetable

def select_teacher(course, timetable):
    available_teachers = []
    for teacher, courses in teachers.items():
        if course in courses and not teacher_exceeds_hours(teacher,
credit_hours[course], timetable):
            available_teachers.append(teacher)

    if not available_teachers:

        return random.choice(list(teachers.keys()))

    return random.choice(available_teachers)



def teacher_exceeds_hours(teacher, credit_hour, timetable):
    # Get the current teaching hours for the teacher
    current_hours = 0
    for day in timetable:
        for class_name, course, t, _, _ in timetable[day]:
            if t == teacher:
                current_hours += credit_hours[course]


    if current_hours + credit_hour > max_teaching_hours:
        return True
```

```
    return False



def select_room():
    return random.choice(available_rooms)

def crossover(parent1, parent2):
    # Create a new timetable for the child
    child_timetable = {}


    for day in working_days:

        crossover_point = random.randint(1, len(parent1[day]))


        child_day_schedule = parent1[day][:crossover_point] +
parent2[day][crossover_point:]


        child_timetable[day] = child_day_schedule

    return child_timetable



def mutation(timetable):
    mutated_timetable = copy.deepcopy(timetable)

    day_scores = calculate_day_scores(mutated_timetable)
    day = roulette_wheel_selection(day_scores)
    timeslot = random.randint(0, len(mutated_timetable[day]) - 1)

    class_details = mutated_timetable[day][timeslot]

    new_teacher = select_teacher(class_details[1], mutated_timetable)  # Pass
mutated_timetable as an argument
    new_room = select_room()
```

```
    mutated_timetable[day][timeslot] = (class_details[0], class_details[1],
new_teacher, new_room, class_details[4])

    return mutated_timetable


def generate_timetable_dataframe(timetable):
    timetable_df = pd.DataFrame(columns=["Class Name", "Course", "Teacher",
"Room", "Timeslot", "Day"])

    i = 0
    for day in working_days:
        for class_details in timetable[day]:
            timetable_df.loc[i] = [class_details[0], class_details[1],
class_details[2], class_details[3], class_details[4], day]
            i += 1

    return timetable_df


best_timetable = genetic_algorithm()

best_timetable_df = generate_timetable_dataframe(best_timetable)

output_file = "C:/Users/Naeem ur rehman/Desktop/fff.csv"

best_timetable_df.to_csv(output_file, index=False)

print(best_timetable_df)
```

Screenshots:

BSCS8B	Neural Ne	Azeema Sa	E 312	11:30:00	thursday
BSCS8B	Software (	Hadiqa Faz	E 409	8:30:00	thursday
BSCS8B	Human Co	Sadia Nazi	E 310	15:30:00	thursday
BSCS8B	Entreprene	Asia Samre	E 315	10:30:00	thursday
BSCS8B	Entreprene	Adnan Ahn	E 314	13:30:00	thursday
BSCS8B	Informatic	Farhanudd	Lab E-218	14:30:00	thursday
BSCS1A	Applied Ca	Asia Samre	E 106	9:30:00	friday
BSCS1A	Applied Ph	Zaryab Qa	E 309	11:30:00	friday
BSCS1A	Computer	Azeema Sa	E 311	14:30:00	friday
BSCS1A	Computer	Azeema Sa	E 311	15:30:00	friday
BSCS1A	Functional	Fasiha Ikra	E 310	12:30:00	friday
BSCS1A	Introducti	Azeema Sa	E 311	9:30:00	friday
BSCS1A	Applied Ph	Ambreen	E 311	14:30:00	friday
BSCS2A	Communic	Laila Nade	E 213	15:30:00	friday
BSCS2B	Communic	Saba Imtia	E 409	9:30:00	friday
BSCS2B	Digital Log	Adnan Ahn	E 310	9:30:00	friday
BSCS2B	Discrete M	Tariq Siddi	E 311	8:30:00	friday
BSCS2B	Object Ori	FL Shahba	Lab E-217	10:30:00	friday
BSCS2B	Object Ori	Adnan Ahn	E 106	14:30:00	friday
BSCS2B	Communic	FL Shahba	E 309	15:30:00	friday
BSCS2B	Communic	Asia Samre	Lab E-218	15:30:00	friday
BSCS3B	Profession	FL Shahba	E 213	13:30:00	friday
BSCS3B	Probability	Azeema Sa	Lab E-218	10:30:00	friday
BSCS2B	Communic	Asia Samre	E 311	15:30:00	friday
BSCS3A	Data Struc	Hadiqa Faz	E 408	13:30:00	friday
BSCS3B	Computer	Aisha Dani	E 213	10:30:00	friday
BSCS3B	Probability	Waqas	E 106	12:30:00	friday
BSCS3B	Probability	Waqas	E 106	8:30:00	friday

BSCS8B	Neural Ne	Azeema Sa	E 312	11:30:00	thursday
BSCS8B	Software (	Hadiqa Faz	E 409	8:30:00	thursday
BSCS8B	Human Co	Sadia Nazi	E 310	15:30:00	thursday
BSCS8B	Entreprene	Asia Samre	E 315	10:30:00	thursday
BSCS8B	Entreprene	Adnan Ahn	E 314	13:30:00	thursday
BSCS8B	Informatic	Farhanudd	Lab E-218	14:30:00	thursday
BSCS1A	Applied Ca	Asia Samre	E 106	9:30:00	friday
BSCS1A	Applied Ph	Zaryab Qa	E 309	11:30:00	friday
BSCS1A	Computer	Azeema Sa	E 311	14:30:00	friday
BSCS1A	Computer	Azeema Sa	E 311	15:30:00	friday
BSCS1A	Functional	Fasiha Ikra	E 310	12:30:00	friday
BSCS1A	Introducti	Azeema Sa	E 311	9:30:00	friday
BSCS1A	Applied Ph	Ambreen	E 311	14:30:00	friday
BSCS2A	Communic	Laila Nade	E 213	15:30:00	friday
BSCS2B	Communic	Saba Imtia	E 409	9:30:00	friday
BSCS2B	Digital Log	Adnan Ahn	E 310	9:30:00	friday
BSCS2B	Discrete M	Tariq Siddi	E 311	8:30:00	friday
BSCS2B	Object Ori	FL Shahba	Lab E-217	10:30:00	friday
BSCS2B	Object Ori	Adnan Ahn	E 106	14:30:00	friday
BSCS2B	Communic	FL Shahba	E 309	15:30:00	friday
BSCS2B	Communic	Asia Samre	Lab E-218	15:30:00	friday
BSCS3B	Profession	FL Shahba	E 213	13:30:00	friday
BSCS3B	Probability	Azeema Sa	Lab E-218	10:30:00	friday
BSCS2B	Communic	Asia Samre	E 311	15:30:00	friday
BSCS3A	Data Struc	Hadiqa Faz	E 408	13:30:00	friday
BSCS3B	Computer	Aisha Dani	E 213	10:30:00	friday
BSCS3B	Probability	Waqas	E 106	12:30:00	friday
BSCS3B	Probability	Waqas	E 106	8:30:00	friday

Ant colony optimization Code

```
import random
import copy
import pandas as pd

# Read the Excel file
file_path = "Rooms.xlsx"

# Read the sheets
rooms_df = pd.read_excel(file_path, sheet_name="Rooms")
teachers_df = pd.read_excel(file_path, sheet_name="teacher")
batch_courses_df = pd.read_excel(file_path, sheet_name="batch courses")
credit_hours_df = pd.read_excel(file_path, sheet_name="credit hours ")
timeslots_df = pd.read_excel(file_path, sheet_name="timeslot")

# Extract the data from the dataframes
available_rooms = rooms_df["Rooms"].tolist()

teachers = {}
for index, row in teachers_df.iterrows():
    teacher_name = row["teacher"]
    course = row["coure"]
    if teacher_name not in teachers:
        teachers[teacher_name] = []
    teachers[teacher_name].append(course)

batch_courses = {}
for index, row in batch_courses_df.iterrows():
    class_name = row["classes"]
    courses = row[1:].dropna().tolist()
    batch_courses[class_name] = courses

credit_hours = {}
for index, row in credit_hours_df.iterrows():
    course = row["coures"]
    credit_hour = row["Credit Hours"]
    credit_hours[course] = credit_hour
```

```python
# Extract the timeslots from the dataframe
timeslots = timeslots_df.set_index("Days").T.to_dict('list')

# Define the working days
working_days = list(timeslots.keys())

# The rest of the code remains the same

def calculate_fitness(timetable):
    fitness_score = 0
    subject_hours = {}
    for day in timetable:
        for class_details in timetable[day]:
            course = class_details[1]
            if course not in subject_hours:
                subject_hours[course] = 0
            subject_hours[course] += 1

    if any(hours > 3 for hours in subject_hours.values()):
        fitness_score -= 1

    if any(hours < 3 for hours in subject_hours.values()):
        fitness_score -= 1

    for day in timetable:
        if day not in working_days:
            fitness_score -= 1

    return fitness_score

def select_teacher(course, timetable):
    available_teachers = []
    for teacher, courses in teachers.items():
        if course in courses and not teacher_exceeds_hours(teacher,
credit_hours[course], timetable):
            available_teachers.append(teacher)

    if not available_teachers:
        return None

    return random.choice(available_teachers)
```

```
def teacher_exceeds_hours(teacher, credit_hour, timetable):
    current_hours = 0
    for day in timetable:
        for class_name, course, t, _, _, _ in timetable[day]:
            if t == teacher:
                current_hours += credit_hours[course]

    if current_hours + credit_hour > max_teaching_hours:
        return True

    return False

def select_room():
    return random.choice(available_rooms)

def initialize_pheromone_trails():
    pheromone_trails = {}
    for day in working_days:
        pheromone_trails[day] = {}
        for class_name, courses in batch_courses.items():
            for course in courses:
                pheromone_trails[day][course] = 1.0  # Initialize pheromone
trails to 1.0 for each course

    return pheromone_trails

from datetime import datetime, timedelta

def construct_timetable(pheromone_trails):
    timetable = {}
    for day in working_days:
        timetable[day] = []

    for class_name, courses in batch_courses.items():
        for course in courses:
            credit_hour = credit_hours[course]
            teacher = select_teacher(course, timetable)
            room = select_room()

            if teacher_exceeds_hours(teacher, credit_hour, timetable):
```

```
                teacher = select_teacher(course, timetable)

            probabilities = []
            total_pheromone = 0.0
            for day in working_days:
                # Check if the class is already scheduled on the same day
                if any(class_name == c[0] for c in timetable[day]):
                    probabilities.append(0.0)  # Set probability to 0 if class is
already scheduled
                else:
                    prob = pheromone_trails[day][course]
                    probabilities.append(prob)
                    total_pheromone += prob

            if total_pheromone == 0.0:
                probabilities = [1.0 / len(probabilities)] * len(probabilities)
            else:
                probabilities = [prob / total_pheromone for prob in
probabilities]

            day_index = roulette_wheel_selection(probabilities)
            day = working_days[day_index]
            timeslot = random.choice(timeslots[day])  # Select a random timeslot
for the selected day

            # Convert the fractional timeslot to a time string
            timeslot = (datetime(1900, 1, 1) +
timedelta(days=timeslot)).time().strftime('%H:%M:%S')

            timetable[day].append((class_name, course, teacher, room, timeslot,
day))

    return timetable

def roulette_wheel_selection(probabilities):
    total_prob = sum(probabilities)
    threshold = random.uniform(0, total_prob)
    cumulative_prob = 0
    for i, prob in enumerate(probabilities):
        cumulative_prob += prob
        if cumulative_prob >= threshold:
```

```
            return i
    return len(probabilities) - 1

def update_pheromone_trails(pheromone_trails, timetable, fitness_score):
    evaporation_amount = evaporation_rate * fitness_score
    for day in timetable:
        for class_name, course, _, _, _, _ in timetable[day]:
            pheromone_trails[day][course] += fitness_score - evaporation_amount

    return pheromone_trails

def ant_colony_optimization():
    best_timetable = None
    best_fitness_score = float("-inf")
    pheromone_trails = initialize_pheromone_trails()

    for _ in range(max_iterations):
        timetable = construct_timetable(pheromone_trails)
        fitness_score = calculate_fitness(timetable)

        if fitness_score > best_fitness_score:
            best_fitness_score = fitness_score
            best_timetable = copy.deepcopy(timetable)

        pheromone_trails = update_pheromone_trails(pheromone_trails, timetable,
fitness_score)

    return best_timetable

def main():
    best_timetable = ant_colony_optimization()
    df = pd.DataFrame(columns=["Class Name", "Course", "Teacher", "Room",
"Timeslot", "Day"])

    for day in working_days:
        for class_details in best_timetable[day]:
            df = df.append({
                "Class Name": class_details[0],
                "Course": class_details[1],
                "Teacher": class_details[2],
                "Room": class_details[3],
```

```
                "Timeslot": class_details[4],
                "Day": class_details[5]
            }, ignore_index=True)

    print(df)

    # Save the DataFrame to a CSV file
    df.to_csv("schedulefinal.csv", index=False)

if __name__ == "__main__":
    main()
```

Screenshots

Class Name	Course	Teacher	Room	Timeslot	Day
BSCS1A	Applied Calculus &	Tooba Mehtab	Lab E-218	11:30:00	monday
BSCS1A	Functional English	Farhanuddin	E 408	11:30:00	tuesday
BSCS1A	Introduction to In	Tahira Afridi	E 313	9:30:00	tuesday
BSCS1A	Computer Progra	Fatima Bashir	Lab E-218	15:30:00	wednesday
BSCS1A	Introduction to In	Talha Alam	E 213	14:30:00	wednesday
BSCS1A	Computer Progra	Azeema Sadia	E 315	15:30:00	thursday
BSCS1A	Applied Physics	Aisha Danish	E 106	12:30:00	friday
BSCS1A	Applied Physics L	Zaryab Qazi	E 313	14:30:00	friday
BSCS1B	Computer Progra	Fatima Bashir	E 315	14:30:00	monday
BSCS1B	Functional English	Farhanuddin	Lab E-219	15:30:00	tuesday
BSCS1B	Introduction to In	Talha Alam	E 106	10:30:00	tuesday
BSCS1B	Applied Physics	Aisha Danish	E 311	12:30:00	wednesday
BSCS1B	Introduction to In	Tahira Afridi	E 310	8:30:00	wednesday
BSCS1B	Applied Physics L	Zaryab Qazi	Lab E-217	8:30:00	thursday
BSCS1B	Computer Progra	Azeema Sadia	Lab E-217	13:30:00	thursday
BSCS1B	Applied Calculus &	Tooba Mehtab	E 409	11:30:00	friday
BSCS1C	Computer Progra	Saba Imtiaz	E 408	8:30:00	monday
BSCS1C	Computer Progra	Azeema Sadia	E 408	15:30:00	tuesday
BSCS1C	Functional English	Farhanuddin	Lab E-218	9:30:00	wednesday
BSCS1C	Applied Calculus &	Tooba Mehtab	E 314	15:30:00	thursday
BSCS1C	Introduction to In	Tahira Afridi	E 310	13:30:00	thursday
BSCS1C	Applied Physics	Aisha Danish	E 408	14:30:00	friday
BSCS1C	Applied Physics L	Zaryab Qazi	Lab E-219	10:30:00	friday
BSCS1C	Introduction to In	Talha Alam	Lab E-217	12:30:00	friday

BSCS2A	Communication S	Shahbano	Lab E-217	11:30:00	monday
BSCS2A	Communication S	Shahbano	Physics Lab	14:30:00	tuesday
BSCS2A	Digital Logic Desig	Waqas	Lab E-217	8:30:00	tuesday
BSCS2A	Foreign Language	Hafiz Harris	E 409	11:30:00	wednesday
BSCS2A	Digital Logic Desig	Dr.Asif	E 409	9:30:00	thursday
BSCS2A	Object Oriented P	Sameena Javed	Lab E-218	9:30:00	thursday
BSCS2A	Object Oriented P	Hafsa Munawer	Lab E-217	8:30:00	thursday
BSCS2A	Discrete Mathem	Amna Iftikhar	E 309	9:30:00	friday
BSCS2B	Communication S	Shahbano	Physics Lab	11:30:00	monday
BSCS2B	Digital Logic Desig	Dr.Asif	Lab E-218	14:30:00	monday
BSCS2B	Foreign Language	Hafiz Harris	E 312	11:30:00	tuesday
BSCS2B	Communication S	Shahbano	Lab E-218	15:30:00	wednesday
BSCS2B	Object Oriented P	Sameena Javed	E 408	11:30:00	wednesday
BSCS2B	Digital Logic Desig	Waqas	E 213	12:30:00	thursday
BSCS2B	Object Oriented P	Hafsa Munawer	E 106	12:30:00	thursday
BSCS2B	Discrete Mathem	Amna Iftikhar	E 408	15:30:00	friday
BSCS3A	Multivariable Cal	Humaira Ahmed	E 309	9:30:00	monday
BSCS3A	Professional Prac	Tooba Zahid	E 414	11:30:00	monday
BSCS3A	Data Structures a	Dr.Raheel	E 409	13:30:00	tuesday
BSCS3A	Data Structures a	Tooba Zahid	Lab E-217	10:30:00	wednesday
BSCS3A	Computer Organi	Aisha Danish	E 106	8:30:00	thursday
BSCS3A	Computer Organi	Ambreen	E 409	13:30:00	thursday
BSCS3A	Probability and St	Jahangir Baig	E 313	13:30:00	friday
BSCS3B	Computer Organi	Aisha Danish	E 213	9:30:00	monday
BSCS3B	Data Structures a	Tooba Zahid	Physics Lab	14:30:00	tuesday
BSCS3B	Multivariable Cal	Humaira Ahmed	Lab E-218	8:30:00	tuesday
BSCS3B	Probability and St	Jahangir Baig	E 408	13:30:00	wednesday
BSCS3B	Computer Organi	Ambreen	Physics Lab	8:30:00	thursday
BSCS3B	Data Structures a	Dr.Raheel	E 314	12:30:00	thursday
BSCS3B	Professional Prac	Tooba Zahid	E 213	8:30:00	friday

BSCS4A	Introduction to P	Fatima Bashir	Lab E-218	11:30:00	monday
BSCS4A	Data Communica	Shaista Ashraf	E 309	9:30:00	tuesday
BSCS4A	Data Communica	Waqas	E 213	14:30:00	wednesday
BSCS4A	Database Management Systems		E 213	14:30:00	wednesday
BSCS4A	Differential Equat	Rizwan Tirmizi	E 408	13:30:00	thursday
BSCS4A	Theory of Autom	Amna Iftikhar	E 213	8:30:00	thursday
BSCS4A	Database Manage	Hafsa Munawer	E 409	11:30:00	friday
BSCS4B	Differential Equat	Rizwan Tirmizi	E 311	12:30:00	monday
BSCS4B	Data Communica	Shaista Ashraf	E 106	9:30:00	tuesday
BSCS4B	Introduction to P	Fatima Bashir	Lab E-219	14:30:00	wednesday
BSCS4B	Data Communica	Waqas	Lab E-217	9:30:00	thursday
BSCS4B	Database Management Systems		Lab E-217	14:30:00	friday
BSCS4B	Database Manage	Hafsa Munawer	Lab E-217	13:30:00	friday
BSCS4B	Theory of Automata		E 311	12:30:00	friday
BSCS5A	Operating Systen	Ambreen	E 414	10:30:00	monday
BSCS5A	Software Enginee	Hadiqa Faza	E 309	15:30:00	monday
BSCS5A	Compiler Constru	Dr.Raheel	E 409	15:30:00	tuesday
BSCS5A	Design and Analy	Sameena Javed	E 213	9:30:00	tuesday
BSCS5A	Operating Systen	Shahid Khan	E 310	11:30:00	wednesday
BSCS5A	Islamic Studies	Abdul Qadir	Lab E-217	8:30:00	thursday
BSCS5A	Linear Algebra	Daniyal Rehman	E 312	12:30:00	thursday
BSCS5A	Compiler Constru	Saba Imtiaz	Lab E-217	10:30:00	friday
BSCS5B	Compiler Constru	Saba Imtiaz	E 106	14:30:00	monday
BSCS5B	Operating Systen	Shahid Khan	E 312	15:30:00	tuesday
BSCS5B	Operating Systems Lab		E 408	8:30:00	tuesday
BSCS5B	Design and Analy	Sameena Javed	E 315	12:30:00	wednesday
BSCS5B	Linear Algebra	Daniyal Rehman	E 414	14:30:00	thursday
BSCS5B	Compiler Construction		E 106	9:30:00	friday
BSCS5B	Islamic Studies		Physics Lab	10:30:00	friday
BSCS5B	Software Enginee	Hadiqa Faza	E 309	14:30:00	friday

Bahria University
Discovering Knowledge

Spring 2023
Department of Computer Science
Bahria University, Karachi Campus

BSCS6A	Introduction to D	Tahira Afridi	E 408	15:30:00	monday
BSCS6A	Web Engineering		E 313	12:30:00	monday
BSCS6A	Numerical Analys	Fasiha Ikram	E 309	13:30:00	tuesday
BSCS6A	Artificial Intelliger	Tahira Afridi	Lab E-217	9:30:00	wednesday
BSCS6A	Technical Writing	Fazal Ali	E 309	11:30:00	thursday
BSCS6A	Web Engineering	Tooba Mehtab	Lab E-217	14:30:00	thursday
BSCS6A	Artificial Intelliger	Tariq Siddique	Lab E-218	11:30:00	friday
BSCS6A	Introduction to D	Laila Nadeem	E 408	12:30:00	friday
BSCS6B	Artificial Intelliger	Tahira Afridi	E 213	9:30:00	monday
BSCS6B	Introduction to D	Laila Nadeem	E 213	9:30:00	tuesday
BSCS6B	Introduction to D	Tahira Afridi	E 312	13:30:00	tuesday
BSCS6B	Technical Writing	Fazal Ali	E 213	12:30:00	tuesday
BSCS6B	Web Engineering	Tooba Mehtab	E 408	13:30:00	tuesday
BSCS6B	Artificial Intelliger	Tariq Siddique	E 309	13:30:00	wednesday
BSCS6B	Numerical Analys	Fasiha Ikram	Physics Lab	10:30:00	thursday
BSCS6B	Web Engineering		E 309	8:30:00	friday
BSCS7B	Organizational Th	M.Awais	E 313	11:30:00	monday
BSCS7B	Natural Language	Asia Samreen	Lab E-219	15:30:00	tuesday
BSCS7B	Pakistan Studies	Asra Naseem	Lab E-218	14:30:00	tuesday
BSCS7B	Parallel & Distribu	Dr.SaifUllah	E 408	9:30:00	wednesday
BSCS7B	Management Infc	Talha Alam	E 213	9:30:00	thursday
BSCS7B	Data Mining		E 408	14:30:00	friday
BSCS7B	Software Project	Malik Ali	Physics Lab	10:30:00	friday
BSCS8A	Human Compute	Ansar Ali	E 414	11:30:00	monday
BSCS8A	InformationÂ Sec	Sadia Nazim	Lab E-218	8:30:00	tuesday
BSCS8A	InformationÂ Sec	Sadia Nazim	E 409	10:30:00	wednesday
BSCS8A	Neural Networks&	Sadia Nazim	E 314	11:30:00	wednesday
BSCS8A	Data Warehousing		E 408	10:30:00	thursday
BSCS8A	Entrepreneurship	Adnan Ahmed	E 315	12:30:00	friday
BSCS8A	Software Quality	Amna Bashir	Lab E-218	15:30:00	friday
BSCS8B	Data Warehousing		E 311	13:30:00	monday
BSCS8B	Entrepreneurship	Adnan Ahmed	E 408	12:30:00	monday
BSCS8B	Human Compute	Ansar Ali	E 408	15:30:00	tuesday
BSCS8B	InformationÂ Security		Lab E-218	13:30:00	tuesday
BSCS8B	Neural Networks& Fuzzy Logic		Lab E-217	9:30:00	wednesday
BSCS8B	InformationÂ Sec	Sadia Nazim	Physics Lab	13:30:00	thursday
BSCS8B	Software Quality	Amna Bashir	Lab E-219	10:30:00	friday

c. **Identify** which algorithms work very well for timetable scheduling problems. [CLO4, PLO4, C3]

ACO is the better algorithm for timetable scheduling problems.

Reasons ACO is the Better Algorithm for Timetable Scheduling Problems:

Robust and Flexible: ACO can handle complex problems with many constraints and objectives without making assumptions about the optimal solution. It explores a wide range of possible solutions, making it robust and flexible in tackling different scheduling scenarios.

Adaptive Nature: ACO adapts its solution approach over iterations based on the information gathered during the search process. It uses pheromone trails to guide the search and updates them dynamically, allowing the algorithm to "learn" and improve its performance over time.

Global Exploration: ACO employs a combination of exploitation (focusing on promising areas) and exploration (searching for new solutions) strategies. This helps the algorithm avoid getting trapped in local optima and increases the chances of finding high-quality solutions.

Scalability: ACO is known for its scalability, making it suitable for large-scale timetable scheduling problems. It can efficiently handle a considerable number of constraints, variables, and resources while maintaining good performance.

Real-World Relevance: ACO has been successfully applied to various real-world scheduling problems, including timetabling. Its effectiveness has been demonstrated in academic institutions, transportation, logistics, and other industries.

In conclusion, ACO is a superior algorithm for timetable scheduling problems due to its robustness, adaptability, global exploration capabilities, scalability, and proven real-world relevance. It can provide high-quality solutions while efficiently handling complex constraints and objectives.

Other algorithms recommended for the task include:

Genetic Algorithms: Easy to implement, flexible, and can provide high-quality solutions. With tuning, they are suitable for solving timetable scheduling problems.

Dynamic Programming: Can find optimal solutions but may face scalability challenges. Worth exploring if the problem can be decomposed into manageable subproblems. Constraint Satisfaction: A viable approach, but it may be computationally expensive for large-scale problems. It could be more applicable to simplified versions of the problem with fewer constraints.

Note: The final choice of algorithm depends on the specific requirements, constraints, and trade-offs of the timetable scheduling problem at hand.

Dataset:

Rooms.xlsx

Rooms
E 408
E 106
Lab E-217
E 409
Lab E-218
E 414
E 106
E 409
E 408
Lab E-218
Lab E-217
Physics Lab
E 213
Lab E-219
E 309
E 310
E 311
E 312
E 313
E 314
E 315

Teachers.xlsx:

teacher	coure
Zaryab Qazi	Physics Lab
Tooba Mehtab	Calculus
Azeema Sadia	Computer Programming (CP)
Talha Alam	IICT
Farhanuddin Raja	ENG
Azeema Sadia	Computer Programming (CP)
Fatima Bashir	Computer Programming Lab
Shahbano	CS
Dr. Asif	DLD
Sameena Javed	OOP
Dr. Asif	Maths II
Hafiz Harris, FL	FL
Amna Iftikhar	DM
FL Shahbano, CS	CS
Dr. Asif	Maths II
Dr. Asif	DLD
Waqas	DLD Lab-Group
Hafsa Munawer	OOP Lab
Tooba Mehtab	Applied Calculus & Analytical Geometry
Aisha Danish	Applied Physics
Azeema Sadia	Computer Programming

	A	B	C	D	E
23	Saba Imtiaz	Computer Programming Lab			
24	**Farhanuddin**	Functional English			
25	Talha Alam	Introduction to Information & Communication Technology			
26	Tahira Afridi	Introduction to Information & Communication Technology Lab			
27	Zaryab Qazi	Applied Physics Lab			
28	Shahbano	Communication Skills			
29	**Dr.Asif**	Digital Logic Design			
30	Waqas	Digital Logic Design Lab			
31	Amna Iftikhar	Discrete Mathematics			
32	Hafiz Harris	Foreign Language			
33	Sameena Javed	Object Oriented Programming			
34	Hafsa Munawer	Object Oriented Programming Lab			
35	Tooba Zahid	Data Structures and Algorithms Lab			
36	Dr.Raheel	Data Structures and Algorithms			
37	Aisha Danish	Computer Organization& Assembly Language			
38	Tooba Zahid	Professional Practices			
39	**Jahangir Baig**	Probability and Statistics			
40	Ambreen	Computer Organization& Assembly Language Lab			
41	Humaira Ahmed	Multivariable Calculus			
42	Malik Ali	Database Management Systems			

43	Shaista Ashraf	Data Communication and Networking		
44	Rizwan Tirmizi	Differential Equations		
45	Waqas	Data Communication & Networking Lab		
46	Fatima Bashir	Introduction to Psychology		
47	Amna Iftikhar	Theory of Automata		
48	Hafsa Munawer	Database Management Systems Lab		
49	**Sameena Javed**	Design and Analysis of Algorithms		
50	Saba Imtiaz	Compiler Construction Lab		
51	**Daniyal Rehman**	Linear Algebra		
52	Hadiqa Faza	Software Engineering		
53	Shahid Khan	Operating Systems		
54	Dr.Raheel	Compiler Construction		
55	Abdul Qadir	Islamic Studies		
56	Ambreen	Operating Systems Lab		
57	Tooba Mehtab	Web Engineering Lab		
58	Fatima Bashir	Web Engineering		
59	**Fasiha Ikram**	Numerical Analysis		
60	Tahira Afridi	Artificial Intelligence Lab		
61	**Tariq Siddique**	Artificial Intelligence		
62	Fazal Ali	Technical Writing & presentation skills		
63	Tahira Afridi	Introduction to Data Science Lab		
64	Laila Nadeem	Introduction to Data Science		
65	M.Awais	Organizational Theory & Behavior		

66	Dr.SaifUllah	Parallel & Distributed Computing
67	**Asia Samreen**	Natural Language Processing
68	Shahid Khan	Data Mining
69	Talha Alam	Management Information System
70	Asra Naseem	Pakistan Studies
71	Malik Ali	Software Project Management
72	Sadia Nazim	Information Security
73	Sadia Nazim	Neural Networks& Fuzzy Logic
74	Amna Bashir	Software Quality Assurance
75	Ansar Ali	Human Computer Interaction
76	Waqas	Data Warehousing
77	**Adnan Ahmed**	Entrepreneurship

Batch course.xlsx

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**classes**	**courses1**	**courses2**	**courses3**	**courses4**	**courses5**	**courses6**	**courses7**	**courses8**	**courses9**			
2	BSCS1A	Applied Calculus & Analytical Geometry	Applied Physics	Computer Programming	Computer Programming Lab	Functional English	Introduction to Information & Communication Technology	Introduction to Information & Communication Technology Lab	Applied Physics Lab				
3	BSCS1B	Applied Calculus & Analytical Geometry	Applied Physics	Computer Programming	Computer Programming Lab	Functional English	Introduction to Information & Communication Technology	Introduction to Information & Communication Technology Lab	Applied Physics Lab				
4	BSCS1C	Applied Calculus & Analytical Geometry	Applied Physics	Computer Programming	Computer Programming Lab	Functional English	Introduction to Information & Communication Technology	Introduction to Information & Communication Technology Lab	Applied Physics Lab				
5	BSCS2A	Communication Skills	Digital Logic Design	Digital Logic Design Lab	Discrete Mathematics	Foreign Language	Object Oriented Programming	Object Oriented Programming Lab	Communication Skills				
6	BSCS2B	Communication Skills	Digital Logic Design	Digital Logic Design Lab	Discrete Mathematics	Foreign Language	Object Oriented Programming	Object Oriented Programming Lab	Communication Skills				
7	BSCS3A	Data Structures and Algorithms Lab	Data Structures and Algorithms	Computer Organization& Assembly Language	Professional Practices	Probability and Statistics	Computer Organization& Assembly Language Lab	Multivariable Calculus					
8	BSCS3B	Data Structures and Algorithms Lab	Data Structures and Algorithms	Computer Organization& Assembly Language	Professional Practices	Probability and Statistics	Computer Organization& Assembly Language Lab	Multivariable Calculus					
9	BSCS4A	Database Management Systems Lab	Data Communication and Networking	Differential Equations	Data Communication & Networking Lab	Introduction to Psychology	Theory of Automata	Database Management Systems					
10	BSCS4B	Database Management	Data Communication	Differential Equations	Data Communication &	Introduction to Psycholo	Theory of Automata	Database Management					

11	BSCS5A	Design and Analysis of Algorithms	Compiler Construction Lab	Linear Algebra	Software Engineering	Operating Systems	Compiler Construction	Islamic Studies	Operating Systems Lab		
12	BSCS5B	Design and Analysis of Algorithms	Compiler Construction Lab	Linear Algebra	Software Engineering	Operating Systems	Compiler Construction	Islamic Studies	Operating Systems Lab		
13	BSCS6A	Web Engineering Lab	Web Engineering	Numerical Analysis	Artificial Intelligence Lab	Artificial Intelligence	Technical Writing & presentation skills	Introduction to Data Science Lab	Introduction to Data Science		
14	BSCS6B	Web Engineering Lab	Web Engineering	Numerical Analysis	Artificial Intelligence Lab	Artificial Intelligence	Technical Writing & presentation skills	Introduction to Data Science Lab	Introduction to Data Science		
15	BSCS7A										
16	BSCS7B	Organizational Theory & Behavior	Parallel & Distributed	Natural Language Processing	Data Mining	Management Information	Pakistan Studies	Software Project Management			
17	BSCS8A	Information Security	Neural Networks & Fuzzy Logic	Software Quality Assurance	Human Computer Interaction	Data Warehousing	Entrepreneurship	Information Security			
18	BSCS8B	Information Security	Neural Networks & Fuzzy Logic	Software Quality Assurance	Human Computer Interaction	Data Warehousing	Entrepreneurship	Information Security			

Credit hour.xlsx

	A	B	C
1	coures	Credit Hou	coure code
2	Applied Calculus & Analytical Geometry	3	GSC-110
3	Applied Physics	2	GSC-114
4	Computer Programming	3	CSC-113
5	Computer Programming Lab	1	CSL-113
6	Functional English	3	ENG-105
7	Introduction to Information & Communication Technology	2	CSC-114
8	Introduction to Information & Communication Technology Lab	1	CSL-114
9	Applied Physics Lab	1	GSL-114
10	Communication Skills	3	HSS-120
11	Digital Logic Design	3	CEN-120
12	Digital Logic Design Lab	1	CEL-120
13	Discrete Mathematics	3	GSC-221
14	Foreign Language	3	HSS-459

27	Data Communication & Networking Lab	4	CEL 222
28	Introduction to Psychology	5	HSS-107
29	Theory of Automata	6	CSC-315
30	Database Management Systems	7	CSC 220
31	Design and Analysis of Algorithms	1	CSC 321
32	Compiler Construction Lab	2	CSL 323
33	Linear Algebra	3	GSC 121

21	Probability and Statistics	3	GSC-122
22	Computer Organization& Assembly Language Lab	1	CEL-324
23	Multivariable Calculus	3	GSC-211
24	Database Management Systems Lab	1	CSL 220
25	Data Communication and Networking	2	CEN 222
26	Differential Equations	3	GSC 210
	Data Communi	4	CEL 222

34	Software Engineering	4	SEN 220
35	Operating Systems	5	CSC 320
36	Compiler Construction	6	CSC 323
37	Islamic Studies	7	ISL 101
38	Operating Systems Lab	8	CSL 320
39	Web Engineering Lab	1	SEL-310
40	Web Engineering	2	SEN-310
41	Numerical Analysis	3	GSC-320
42	Artificial Intelligence Lab	1	CSL-411
43	Artificial Intelligence	3	CSC-411

43	Artificial Intelligence	3	CSC-411
44	Technical Writing & presentation skills	3	HSS-320
45	Introduction to Data Science Lab	1	CSL-487
46	Introduction to Data Science	2	CSC-487
47	Organizational Theory & Behavior	3	MGT 242
48	Parallel & Distributed Computing	3	CEN 455
	Natural	3	CSC 441

49	Natural Language Processing	3	CSC 441		
50	Data Mining	3	CSC 452		
51	Management Information System	3	CSC 458		
52	Pakistan Studies	2	PAK 101		
53	Software Project Management	3	SEN 410		
54	Information Security	3	CSC 407		
55	Neural Networks & Fuzzy Logic	3	CSC 449		

56	Software Quality Assurance	3	SEN 420
57	Human Computer Interaction	2	SEN 320
58	Data Warehousing	3	CSC 454
59	Entrepreneurship	3	HSS 410

Spring 2023
Department of Computer Science
Bahria University, Karachi Campus

Timeslots.xlsx

Days	Time Slot 1	Time Slot 2	Time Slot 3	Time Slot 3	Time Slot 4	Time Slot 4	Time Slot 5	Time Slot 6
monday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
tuesday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
wednesd	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
thursday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
friday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
monday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
tuesday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
wednesd	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
thursday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
friday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
monday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
tuesday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
wednesd	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
thursday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
friday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
monday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
tuesday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
wednesd	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
thursday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
friday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
monday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
tuesday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
wednesd	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
thursday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
friday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
monday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
tuesday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
wednesd	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
thursday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
friday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
monday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
tuesday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
wednesd	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
thursday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
friday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
monday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
tuesday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
wednesd	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
thursday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00
friday	8:30:00	9:30:00	10:30:00	11:30:00	12:30:00	13:30:00	14:30:00	15:30:00