# Assignment-1: Conditional Statements

---

**1.Student Grading System**

- **Task**: Create a program to calculate a student's grade based on their marks.
- **Input**: Prompt the user to enter their marks.
- **Grading Criteria**:
  - Grade A: 90–100
  - Grade B: 80–89
  - Grade C: 70–79
  - Grade D: 60–69
  - Grade E: 50–59
  - Grade F: 0–49
  - Invalid Marks: Outside the range 0–100.
- **Output**: Display the grade or an error message for invalid marks.

---

## 2. Tax Calculation for Car Purchase

Write a  program to calculate the **tax** on a car purchase based on the car brand and its price.

2. **Mahindra**: 5% tax for prices between 7L (7 lakh) and 10L.
3. **Audi**: 10% tax for prices between 10L and 15L.
4. **Jaguar**: 25% tax for prices between 15L and 20L.
5. **Mercedes**: 30% tax for prices between 20L and 25L.
6. Input: The car brand and price.
7. Output: The calculated tax on the purchase.

---

**3.Finding the Middle Number**

- **Task**: Write a program to determine the middle number among three inputs.
- **Input**: Prompt the user to enter three numbers.
- **Processing**: Identify the middle number, which is neither the largest nor the smallest.
- **Output**: Display the middle number.

---

**4.Find the greatest number.**

- **Task**: Write a program to find greatest number from three number
- **Input**: Prompt the user to enter three numbers.
- **Output**: Display the greatest number.

---

**5.Authentication System**

- **Task**: Write a program to authenticate a user by validating their username and password.
- **Predefined Credentials**:
  - Username: `user1`
  - Password: `pass@123`
- **Input**: Prompt the user to input their username and password.
- **Output**:
  - If the credentials match, display "Authentication successful."
  - If they do not match, display "Authentication failed."

---

**6.Calculate Class Attendance Percentage**

- **Task**: Write a program to calculate the percentage of classes attended by a student and determine their eligibility to sit in the exam.
- **Conditions**:
  - Attendance percentage < 75%: Not eligible to sit in the exam.
  - Attendance percentage ≥ 75%: Eligible to sit in the exam.
- **Output**: Display the attendance percentage and eligibility status.

---

**6.Library Charge Calculation**

- **Task**: Write a program to calculate the library charges based on the number of days a book has been borrowed.
- **Charge Criteria**:
  - Up to 5 days: ₹2/day.
  - 6 to 10 days: ₹3/day.

        - 11 to 15 days: ₹4/day.
        - More than 15 days: ₹5/day.
    - **Output**: Display the total charges.

---

**8.UPSC Selection Process**

- **Task**: Simulate the UPSC selection process with the following steps:

    1. **Eligibility Check**

        - **Criteria**:
            - Age: 21–32 years.
            - Graduate status: Must be a graduate.
            - Nationality: Must be "Indian".
        - **Output**:
            - If eligible, proceed to Prelims.
            - If ineligible, display the reason for ineligibility.
    2. **Prelims Exam**

        - **Processing**: Check if the candidate's score ≥ cut-off.
        - **Output**:
            - If passed, proceed to Mains.
            - If failed, display "You failed the Prelims."
    3. **Mains Exam**

        - **Processing**: Check if the candidate's score ≥ cut-off.
        - **Output**:
            - If passed, proceed to Interview.
            - If failed, display "You failed the Mains."
    4. **Interview**

        - **Processing**: Check if the candidate's score ≥ cut-off.
        - **Output**:
            - If passed, display "Congratulations! You have cleared the UPSC."
            - If failed, display "You failed the Interview."
- **Final Output**: Use nested conditional statements to simulate the entire process.

---

## 9. **Menu-Driven Login System**

1. **Create the Menu:**
    - Display a menu with three choices for the user:
        - Login with Phone
        - Login with Email
        - Exit the system
2. **Predefined Credentials:**
    - Phone number: `"1234567890"`
    - OTP: `"1234"`
    - Email: `"user@example.com"`
    - Password: `"password123”`
3. **Login Functionality:**
    - **Option 1 (Login with Phone):**
        - Prompt the user to enter their phone number and OTP.
        - Compare the input with a predefined phone number and OTP.
        - Display success if both match or an error message if they don’t.
    - **Option 2 (Login with Email):**
        - Prompt the user to enter their email and password.
        - Compare the input with predefined email and password.
        - Display success if both match or an error message if they don’t.
    - **Option 3 (Exit):**
        - Display an exit message and terminate the program.
    - **Invalid Input:**
        - Handle invalid user choices and ask the user to select a valid option.

### Output:

1. If the user enters a valid phone number and OTP, display: `"Login successful with phone!"`
2. If the user enters valid email and password, display: `"Login successful with email!"`
3. If the user selects the exit option, display: `"Exiting the program. Have a nice day!"`
4. If the user enters invalid credentials or an invalid choice, display appropriate error messages.

---

## 10.Create Your Own KBC Game

Design and implement a quiz game inspired by the popular *Kaun Banega Crorepati (KBC)* game show. The aim of this assignment is to test the user's knowledge through a series of multiple-choice questions, track their score, and display statistics at the end of the game. The game also provides the flexibility to skip any question.

## Instructions:

1. **Game Structure:**
    - The game will consist of **5 multiple-choice questions**.
    - The user will be asked a question with **4 options** (A, B, C, D).
    - The user can choose to **skip** any question they do not want to answer.
2. **Scoring System:**
    - Points will be awarded for **correct answers** as follows:
        - **Question 1** → 1000 points
        - **Question 2** → 2000 points
        - **Question 3** → 3000 points
        - **Question 4** → 5000 points
        - **Question 5** → 10000 points
    - For **incorrect answers**, no points will be awarded.
    - For **skipped questions**, no points will be awarded, but the game will continue.
3. **End of Game Statistics:**
    - At the end of the game, the following **statistics** will be displayed:
        - **Total score** accumulated from correct answers.
        - **Number of correct answers** provided by the user.
        - **Number of skipped questions**.
        - **Number of wrong answers**
4. **User Experience:**
    - At the beginning of the game, ask the user whether they would like to **start** the game.
    - Provide the option for the user to **skip** any question at any point.
    - After the game is over, ask the user if they would like to **rate** the game (e.g., with a 1-5 star rating) to gauge user feedback.
5. **Game Ending:**
    - The game will end when all the questions have been answered or skipped. The user should receive their total score and a summary of their performance.