

1. Login Authentication System

- Create a UserLogin class.
- Ask the user for a username and password.
- Hardcode a correct username & password (admin, pass123).
- If the user enters incorrect details 3 times, exit the program.
- Use exception handling for input mismatches.

2. Random Number Program

- Generate a random number between 1-100.
- Ask the user to guess the number.
- Provide hints like:
 - "Too high" if the guess is greater.
 - "Too low" if the guess is smaller.
- If the user guesses correctly, congratulate them and exit.
- Handle invalid inputs (non-numeric values).

3. ATM Transaction System

- Create an ATM class.
- Ask the user for a PIN (hardcoded).
- Provide the following options:
 - Check Balance
 - Withdraw Money (Ensure sufficient balance)
 - Deposit Money (Ensure deposit is > 0)
 - Exit
- Use exception handling for invalid inputs.

4. Railway Ticket Booking (Without Data Structures)

- Create a Passenger class with:
 - Name
 - Age
 - Ticket Status (Booked/Not Booked)
- Ask the user to enter their details.
- If age < 5, deny booking.
- If age > 60, apply a discount.
- Confirm ticket booking.
- Handle invalid age inputs.

5. Custom Exception Handling for Banking

- Modify the Banking System.
- Create custom exceptions:
 - InsufficientBalanceException
 - NegativeAmountException
- Ensure:
 - Users cannot withdraw more than balance.
 - Users cannot enter negative amounts.
 - Login first and then user can perform other functionalities
 - Number of login attempts can be 3
 - Use encapsulated class for login accno and password

6. Employee Payroll System

- Create an Employee class with:
 - Name
 - Basic Salary
- Calculate Net Salary with:
 - HRA = 10% of Basic Salary
 - DA = 5% of Basic Salary
 - Tax = 10% if salary > 50,000
- Display salary details.
- Handle negative salary input exceptions.

7. Password Strength Checker

- Ask the user to enter a password.
- Check if it meets the following conditions:
 - At least 8 characters long.
 - Contains at least one uppercase letter.
 - Contains at least one digit.
 - Contains at least one special character (@, #, \$ etc.).
- Display "Weak", "Medium", or "Strong" based on the conditions.

8. Simple Chatbot (Using Strings & Control Statements)

- Create a chatbot simulation.
- The chatbot should respond based on user input:
 - If the user types "hi" → "Hello! How can I help you?"
 - If the user types "bye" → "Goodbye!" and exit.
 - If the user types anything else → "I don't understand. Try again."
- Continue the conversation until "bye" is entered.

9. Online Exam System

- Create a console-based Online Exam System where a user can:
 1. Login using a username and password (Hardcoded credentials).
 2. Take an Exam (Answer 5 multiple-choice questions).
 3. View Results (Display score after completing the exam).
 4. Logout & Exit.