	School of Engineering & Technology	
	Department: CSE	**Session:** Odd
	Programme: B.Tech CSE (Specialization in AI & ML)	**Semester:** 1
	Course Code: ETCCCP105	**Number of students: VARUN GAUR**
	Course Name: Computer Science Fundamentals & Career Pathways	**Faculty: Dr. Ravinder Beniwal**
KRMU	**Roll No**: 2501730029	**Date**: October 2025

Assignment Number 01: Foundations of Computer Science & Computational Thinking

Assignment Title: *Design and Simulate a Real-World Process Using Flowcharts and Pseudocode*

ATM Withdrawal Process

The ATM (Automated Teller Machine) withdrawal process is a real-world example of computational thinking. It involves authentication, transaction selection, balance checking, and record updating—each representing clear input, processing, and output stages.

Input, Process, and Output:

Stage	Description
Input	Card details, PIN number, withdrawal amount
Process	Verify PIN → Check balance → Validate amount → Dispense cash → Update account
Output	Cash dispensed, updated balance, transaction receipt

Problem Analysis

1. Abstraction:

Essential steps include inserting the card, verifying PIN, entering amount, checking balance, dispensing cash, and updating the account. Irrelevant details like hardware or network latency are ignored.

2. Decomposition

The ATM withdrawal process can be divided into the following sub-tasks:

1. **Insert Card & Enter PIN**
→ Capture user card and read data.
2. **PIN Verification**
→ Compare entered PIN with stored PIN.
3. **Amount Input & Validation**
→ Ask for amount and ensure it’s within daily limit and balance.
4. **Transaction Processing**
→ Deduct amount from balance and update account records.
5. **Cash Dispensing**
→ Dispense the requested amount.
6. **Receipt Generation & Exit**
→ Print transaction receipt and eject card.

3. Pattern Recognition

The process of **input → validation → processing → output** appears in many systems:

- Online shopping payments
- Mobile banking transfers
- Railway ticket booking

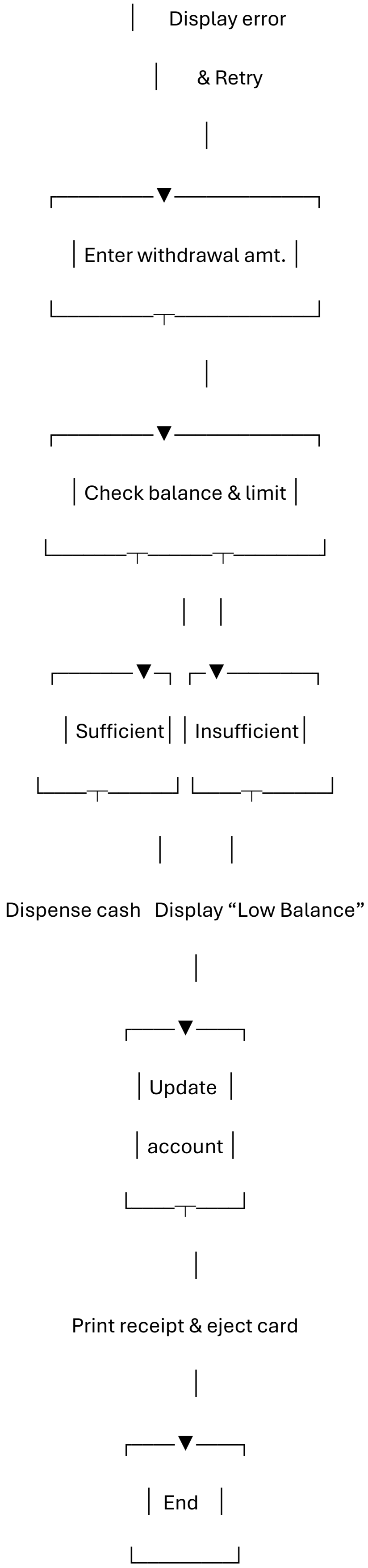
Solution Design

(a) Flowchart

Flowchart Description:

Start → Insert Card → Enter PIN → Verify PIN → Valid? → Enter Amount → Check Balance → Sufficient? → Dispense Cash → Update Account → Print Receipt → End

```
┌────────────────────┐
│ Start              │
└────────┬───────────┘
         │
┌────────▼───────────┐
│ Insert card & Enter PIN│
└────────┬───────────┘
         │
┌────────▼───────────┐
│ Verify PIN         │
└─────┬───────┬──────┘
      │       │
┌─────▼─┐ ┌─▼──────┐
│ Valid │ │ Invalid │
└──┬────┘ └──┬─────┘
   │         │
```



Display error

& Retry

Enter withdrawal amt.

Check balance & limit

Sufficient

Insufficient

Dispense cash

Display “Low Balance”

Update

account

Print receipt & eject card

End

(b) Pseudocode

```
BEGIN
  DISPLAY "Welcome to ATM"
  INSERT card
  ENTER pin
  IF pin == stored_pin THEN
    DISPLAY "PIN Verified"
    ENTER amount
    IF amount <= balance THEN
      balance = balance - amount
      DISPENSE cash
      PRINT "Transaction Successful"
      PRINT "Remaining Balance:", balance
    ELSE
      PRINT "Insufficient Balance"
    ENDIF
  ELSE
    PRINT "Invalid PIN. Please Try Again."
  ENDIF
  EJECT card
  DISPLAY "Thank You for Using ATM"
END
```

Implementation (Python Code)

```
#ATM Withdrawal Simulation

#Author: Varun Gaur

#Purpose: Simulate ATM withdrawal process
```

```
#Stored account details (for simulation)

stored_pin = 1234

balance = 5000 # Initial balance

#Step 1: Input

print("Welcome to the ATM Machine") entered_pin = int(input("Enter your 4-digit PIN: "))

#Step 2: PIN Verification

if entered_pin == stored_pin: print("PIN Verified ✅") amount = int(input("Enter withdrawal amount: ₹"))

# Step 3: Check balance and process
if amount <= balance and amount > 0:
    balance -= amount
    print(f"Please collect your cash: ₹{amount}")
    print(f"Remaining Balance: ₹{balance}")
    print("Transaction Successful ✅")
else:
    print("❌ Insufficient Balance or Invalid Amount")


else: print("❌ Invalid PIN. Please Try Again.")

print("Card Ejected. Thank You for Using ATM 💳")
```

Code Explanation:

- The program stores a predefined **PIN** and **balance**.
- It asks the user for a PIN and verifies it.
- If valid, it proceeds to **withdrawal and balance check**.
- If the transaction is successful, it updates and displays the new balance.
- Error messages are displayed for invalid PIN or insufficient balance.

Reflection

Challenges Faced:

- Structuring the pseudocode and ensuring logical flow.
- Managing conditions for invalid inputs.
- Ensuring user-friendly messages for each step.

Insights Gained:

- Learned how **computational thinking** simplifies complex real-world processes.
- Understood how abstraction and decomposition improve program clarity.

Potential Improvements:

- Add support for multiple transactions.
- Include deposit and balance inquiry options.
- Implement security features like limited PIN attempts.

Conclusion

The ATM withdrawal process is a perfect demonstration of computational thinking in action. By breaking the process into smaller steps, identifying patterns, and using abstraction, we can design an efficient and reliable solution — from flowchart to working Python code.