# 🧠 **Assignment: Build a Single Layer Perceptron Using Python Basics**

## 🎯 Objective

Build a **Single Layer Perceptron from scratch** using **only Python basics** (no ML libraries) to solve a **binary classification problem** of your own design.

---

## 📌 Problem Statement

Create a system that predicts whether a student will **Pass (1)** or **Fail (0)** based on:

- Study hours per day
- Class attendance percentage

⚠️ You must **create your own unique dataset**. Using online datasets or ChatGPT-generated data is not allowed.

---

## 🔢 Input Format

Each data point should be in the form:

[study_hours, attendance_percentage]

Example (format only, do not copy):

[4, 65]

---

## 🎯 Output

- `1` → Pass
- `0` → Fail

---

## 🛠️ Mandatory Rules

### ❌ Not Allowed

- numpy
- sklearn
- tensorflow / keras/ pytorch
- Copying full code from ChatGPT or classmates

✅ Must Use

- `for` loops
- `if-else` conditions
- `lists`
- `functions`
- user input (`input()`)
- meaningful `print()` output

---

⚙️ Task Breakdown

Task1️⃣: Dataset Creation

- Create **10–15 data points**
- Store features in a list of lists
- Store labels in a separate list

💡 Labels must be assigned using **your own rule** (clearly explained in the code comment).

---

Task2️⃣: Perceptron Initialization

Manually define:

- Two weights
- One bias
- Learning rate

You may choose random or fixed values, but you must justify your choice.

Task3️⃣: Activation Function

Write a function that:

- Calculates the weighted sum
- Applies a threshold of your own choice(step function)
- Returns `0` or `1`

---

Task4️⃣: Training Loop

- Train the perceptron for multiple epochs

- For each data point:
  - Predict the output
  - Calculate the error
  - Update weights and bias

📌 The **weight update rule must be written and explained by you** (in comments).

---

### Task 5️⃣: User Input Testing

After training:

- Take study hours and attendance as user input
- Predict pass/fail using the trained model
- Print a meaningful message
  - Example: *“The student is likely to Pass”*

---

### 📝 Short Report (Required)

Students must submit a brief explanation covering:

1. How the dataset was created
2. Why the learning rate was chosen
3. How they verified the model is learning

⚠️ Identical datasets or explanations will be treated as plagiarism.

---

### 🧩 Bonus (Optional)

- Calculate accuracy
- Print training loss per epoch
- Show how weights change over time