

Minor in AI

20 March 2025

Revision - Mastering Input and Decision-Making in Python: From Basics to Logic

```
# Variables
# age (int)
# height (int, in cm)
# is_accompanied (bool)
# is_vip (bool)
# has_heart_condition (bool)
# is_banned (bool)

def can_enter_ride():
    if has_heart_condition:
        print("Entry Denied: Health risk due to heart condition.")
    elif is_vip and not is_banned:
        print("Entry Allowed: VIP access granted.")
    elif is_vip and is_banned:
        print("Entry Denied: VIP banned from rides.")
    elif age >= 12 and height >= 140:
        if 12 <= age <= 15 and not is_accompanied:
            print("Entry Denied: Ages 12-15 must be accompanied.")
        else:
            print("Entry Allowed: Meet age and height requirements.")
    else:
        print("Entry Denied: Does not meet age or height requirements.")

age = 13
height = 145
is_accompanied = True
is_vip = False
has_heart_condition = True
is_banned = False
can_enter_ride()
```

```

import random
import matplotlib.pyplot as plt
# Initialize counters and totals
low_count = 0
low_total = 0
average_count = 0
average_total = 0
high_count = 0
high_total = 0
scores = []

# Generate and process 100 random numbers
for i in range(100):
    num = random.randint(1, 100)
    score = num // 10
    scores.append(score)

    if score <= 4:
        low_count += 1
        low_total += score
    elif score <= 7:
        average_count += 1
        average_total += score
    else:
        high_count += 1
        high_total += score

if low_count > 0:
    low_mean = low_total / low_count
else:
    low_mean = 0

if average_count > 0:
    average_mean = average_total / average_count
else:
    average_mean = 0

if high_count > 0:
    high_mean = high_total / high_count
else:
    high_mean = 0

# Simple prints
print("Low Bucket (0-4):", low_count, "scores, Mean =",
      round(low_mean, 2))
print("Average Bucket (5-7):", average_count, "scores, Mean =",
      round(average_mean, 2))
print("High Bucket (8-10):", high_count, "scores, Mean =",
      round(high_mean, 2))

plt.scatter(range(100), scores, color='blue')
plt.title("Scatter Plot of Scores")
plt.show()

```