



Homework Assignment

Week 8: Multiple Paths with if/else and elif

Total Points: 100 — Due: Next Class



Name: _____

Date: _____

Important Instructions

- Work on your own, but you may ask family members for hints only if stuck
- Write your code in Python (*using Thonny or any online IDE*)
- For setting up your code files, kindly see [instructions.pdf](#)
- For written answers, use the spaces provided on this sheet
- Show your work for full credit!

1 Part 1: Simple if/else Practice (15 points)

Time estimate: 20 minutes

1.1 Exercise 1.1: Exam Pass/Fail Checker (7 points)

Open [week8_ex1.1.py](#) and complete this simple program that tells the students if they passed:

```
1 # Simple Pass/Fail Checker
2 print("=== Exam Result Checker ===")
3
4 marks = int(input("Enter your marks out of 100: "))
5 passing_marks = 40
6
7 # Check if student passed
8 if _____ >= _____:
9     print("Congratulations! You passed!")
10    print(f"You scored {marks - passing_marks} marks above passing.")
11    _____:
12    print("Sorry, you did not pass.")
13    print(f"You need {passing_marks - marks} more marks to pass.")
14
15 print("Keep working hard!")
```

1.2 Exercise 1.2: Temperature Comfort Zone (8 points)

Open [week8_ex1.2.py](#) and complete this temperature comfort checker. It should give advice based on the temperature input:

```

1 # Temperature Comfort Checker
2 temperature = float(input("What's the temperature in Celsius? "))
3
4 # Simple hot or cold check
5 ----- > 30:
6     print("It's hot today!")
7     print("Drink plenty of water.")
8 -----:
9     print("Temperature is comfortable.")
10    print("Enjoy your day!")
11
12 # Another check for cold
13 ----- < 15:
14     print("It's quite cold!")
15     print("Wear warm clothes.")
16 -----:
17     print("Not too cold.")
18
19 # Check for extreme temperatures
20 if temperature > 40 ----- temperature < 5: # Fill in
21     print("WARNING: Extreme weather conditions!")

```

2 Part 2: Working with elif (18 points)

Time estimate: 25 minutes

2.1 Exercise 2.1: Game Level Progression (10 points)

Open [week8_ex2_1.py](#) and complete this game level progression system using an `elif` chain. The player should receive a level based on their experience points (XP):

XP Range	Level
0-99	Beginner
100-299	Intermediate
300-599	Advanced
600-999	Expert
1000 and above	Master

Table 1: Game Progression Chart

```

1 # Game Level Progression System
2 print("=== Player Level Calculator ===")
3 xp_points = int(input("Enter your experience points (XP): "))
4
5 # Determine player level using elif chain
6 if ----- :
7     level = "Master"
8     bonus = "Unlock legendary weapons!"
9 elif ----- :
10    level = "Expert"
11    bonus = "Access to expert quests!"
12 ----- :
13    level = "Advanced"
14    bonus = "Can join guild battles!"
15 ----- :
16    level = "Intermediate"
17    bonus = "New skills available!"
18 ----- :

```

```

19     level = "Beginner"
20     bonus = "Keep playing to level up!"
21
22     print(f"\nYour level: {level}")
23     print(f"Status: {bonus}")
24
25     # Check if close to next level
26     xp_to_next = 0
27     if level == "Beginner":
28         xp_to_next = 100 - xp_points
29     ----- level == "Intermediate":           # Fill in: elif
30         xp_to_next = 300 - xp_points
31     # Add more checks...
32
33     if xp_to_next > 0 ----- xp_to_next <= 20: # Fill in: and
34         print(f"So close! Just {xp_to_next} XP to next level!")

```

Save this code in `week8_ex2.1.py`

2.2 Exercise 2.2: Social Media Post Timing (8 points)

Fix the order and complete this social media scheduler:

Order	Code Line
--	elif followers < 1000:
line 1	print("=== Smart Post Scheduler ===")
--	print("Growing account - post 2x daily")
--	print("Build your audience strategically!")
--	followers = int(input("Number of followers: "))
--	print("New account - post daily to grow!")
--	if followers < 100:
--	elif followers < 10000:
--	print("Established - post 3-4x weekly")
--	else:
--	print("Influencer! Quality over quantity")

3 Part 3: Nested if Statements (15 points)

Time estimate: 20 minutes

3.1 Exercise 3.1: Movie Ticket Pricing (15 points)

The following code calculates movie ticket prices based on age, day, and student status.

Movie Theater Rules

- Regular ticket: Rs. 300
- Children (under 12): Rs. 150
- Students get 20% off their ticket type
- Wednesday is discount day: All tickets Rs. 200

Open [week8_ex3.1.py](#) and complete the nested conditions for the movie ticket calculator:

```

1 # Movie Ticket Calculator with Nested Conditions
2 print("=== Movie Ticket Price Calculator ===")
3
4 day = input("What day is it? ")
5 age = int(input("Enter your age: "))
6
7 # First check if it's discount Wednesday
8 if day == "Wednesday" :
9     price = 200
10    print("Wednesday special - all tickets Rs. 200!")
11 else:
12     # Not Wednesday, so check age for pricing
13     ----- age < 12:
14         price = 150
15         print("Child ticket applied.")
16     -----:
17         price = 300
18         print("Regular ticket price.")
19
20     # Now check if they're a student (nested condition)
21     if age >= 12 and age <= 25:
22         is_student = input("Are you a student? (yes/no): ")
23         ----- is_student == "yes":
24             # Apply 20% student discount
25             discount = ----- * 0.2
26             price = price - discount
27             print(f"Student discount applied! Saved Rs. {discount}")
28
29 print(f"\nTotal ticket price: Rs. {price}")
30
31 # Extra service check
32 if price <= 200:
33     print("Great deal! Enjoy the movie!")
34     ----- day == "Wednesday": # Fill in: if
35         print("Thanks for choosing our Wednesday special!")

```

Save your progress in `week8_ex3.1.py`

4 Part 4: Complex Conditions with and/or (18 points)

Time estimate: 25 minutes

4.1 Exercise 4.1: Sports Team Eligibility (10 points)

Open `week8_ex4.1.py` and complete this sports team selection system using `and/or` operators. The program should check multiple conditions for eligibility:

```

1 # Sports Team Selection System
2 print("=== School Cricket Team Selection ===")
3
4 age = int(input("Enter your age: "))
5 fitness_score = int(input("Fitness score (1-10): "))
6 skill_level = int(input("Cricket skill level (1-10): "))
7 attendance = int(input("Practice attendance percentage: "))
8
9 # Check basic eligibility (must meet ALL requirements)
10 if age >= 12 ----- age <= 17 ----- attendance >= 80: # Fill
11     operators
12     print("Basic requirements met!")

```

```

12
13 # Check performance criteria (need high skill OR fitness)
14 if skill_level >= 8 _____ fitness_score >= 8:           # Fill
15     operator
16     print("Selected for A Team!")
17     # Extra check for captain eligibility
18     if skill_level >= 9 _____ fitness_score >= 9:       # Fill
19         operator
20         print("Eligible for team captain consideration!")
21 elif skill_level >= 6 _____ fitness_score >= 6:       # Fill
22     operators
23     print("Selected for B Team!")
24 else:
25     print("Keep practicing to improve your scores!")
26 else:
27     print("Does not meet basic requirements.")
28     # Helpful feedback
29     if age < 12 _____ age > 17:                         # Fill operator
30         print("- Age must be between 12 and 17")
31     if attendance < 80:
32         print("- Attendance must be at least 80%")

```

4.2 Exercise 4.2: Debug Multiple Small Programs (8 points)

Each snippet has issues. Without running the code, find and fix all. circle the issues on paper and write the (*just the*) corrected code by hand to the right of each snippet.

Snippet A: Password Checker (2 points)

```

1 # Syntax Error - Fix it!
2 password = input("Enter password: ")
3 If password == "secret123":
4     print("Access granted!")
5 else:
6     print("Access denied!")

```

Snippet B: Age Group (2 points)

```

1 # Logic Error - This will never print "Teenager"!
2 age = 15
3 if age < 20:
4     print("Young person")
5 elif age >= 13 and age <= 19:
6     print("Teenager")
7 else:
8     print("Adult")

```

Snippet C: Weekend Checker (2 points)

```

1 # Logic Error - Why does Tuesday count as weekend?
2 day = input("What day is it? ")
3 if day == "Saturday" or "Sunday":
4     print("It's the weekend!")
5 else:
6     print("It's a weekday.")

```

5 Part 5: Code Tracing Challenge (12 points)

Time estimate: 15 minutes

5.1 Exercise 5.1: Trace the Pizza Order (12 points)

For this exercise you are not allowed to run the code. You must trace it manually in your head. Read this code carefully and write exactly what it prints:

```

1 # Pizza Order System
2 size = "medium"
3 toppings = 3
4 day = "Friday"
5
6 # Calculate base price
7 if size == "small":
8     price = 100
9 elif size == "medium":
10    price = 150
11 else:
12    price = 200
13
14 print(f"Base price for {size} pizza: Rs. {price}")
15
16 # Add topping charges
17 if toppings > 2:
18     extra = (toppings - 2) * 30
19     price = price + extra
20     print(f"Extra toppings charge: Rs. {extra}")
21
22 # Check for weekend deal
23 if day == "Friday" or day == "Saturday":
24     if price > 150:
25         discount = 25
26         price = price - discount
27         print("Weekend special: Rs. 25 off!")
28     else:
29         print("Weekend deal applies to orders over Rs. 150")
30
31 print(f"Final price: Rs. {price}")
    
```

Write the exact output below (what the program prints):

1: _____

2: _____

3: _____

4: _____

Tracing Questions (3 points each):

1. What is the value of price after the size check? _____
2. Does the extra toppings charge get added? (yes/no) _____
If yes, what is the new price? _____
3. Does the weekend special discount apply? (yes/no) _____
Why or why not? _____
4. What would the final price be if day = "Monday"? _____

6 Part 6: Program Development (22 points)

Time estimate: 35 minutes

Open [week8_ex6.py](#) and complete the following program that gives personalized health advice based on user input. Follow the requirements below:

Program Requirements

Your program should:

- Ask for user's age
- Ask about daily water intake (glasses)
- Ask about sleep hours
- Ask about exercise (yes/no and minutes if yes)
- Ask about screen time (hours)
- Provide health score (0-100) based on:
 - Water: 8+ glasses = full points (20)
 - Sleep: Age-appropriate hours = full points (25)
 - Exercise: 30+ minutes = full points (25)
 - Screen time: Under limit = full points (30)
- Give specific advice using multiple elif chains
- Use nested conditions for age-specific recommendations
- Use and/or for complex health warnings

Age-specific guidelines:

- Ages 6-12: 9-11 hours sleep, max 2 hours screen
- Ages 13-17: 8-10 hours sleep, max 3 hours screen
- Ages 18+: 7-9 hours sleep, max 4 hours screen

Example Program Run:

```

=== Daily Health Tracker ===
Enter your age: 14
Glasses of water today: 6
Hours of sleep last night: 7
Did you exercise today? (yes/no): yes
How many minutes? 45
Hours of screen time: 4

--- Health Report Card ---
Health Score: 73/100

Water Intake: 6 glasses
- Below recommended 8 glasses (-7 points)
- Try to drink 2 more glasses daily!

Sleep: 7 hours
- Below recommended 8-10 hours for your age (-6 points)
- Teenagers need more sleep for growth!
    
```

Exercise: 45 minutes

- Excellent! Full points (25/25)
- Keep up the great work!

Screen Time: 4 hours

- Over the 3-hour limit for your age (-9 points)
- Reduce by 1 hour for better health

Overall Assessment: Good effort!

Priority: Improve sleep and reduce screen time.

Remember: Your health habits today shape your future!

Testing Checklist:

- Program runs without errors
- Score calculation is accurate
- Age-specific advice works correctly
- All four health categories are evaluated
- Uses if/elif/else chains effectively
- Includes nested conditions
- Uses at least one and/or combination
- Provides helpful, specific advice

7 Part 7: Reflection Questions (5 points)

Time estimate: 10 minutes

Answer these questions in complete sentences:

1. Think about the elif chains you wrote. Why is the order important? What would happen if you checked for a grade of 60 before checking for 90?

2. When you use nested if statements, the inner if only runs when the outer if is True. Give a real-life example where this makes sense (not from the homework).

3. Look at this condition: `if age >= 13 and age <= 19`. Explain what values make this True. Can you think of another way to write this?

8 Bonus Section: Extra Challenges (Optional - 10 extra points)

Only attempt if you've finished everything else!

8.1 Challenge 1: Password Strength Analyzer (5 points)

Open [week8_ex8.1.py](#) and create a password strength checker using the following requirements:

Password Requirements

Your program should:

1. Ask the user to enter a password
2. Determine the strength based on length:
 - Less than 6 characters = *weak*
 - 6 to 10 characters = *medium*
 - More than 10 characters = *strong*
3. Check if the password is one of three common bad passwords: "password", "123456", or "admin"
4. Use **nested if statements** to give different messages based on BOTH strength and whether it's a bad password

Program output rules:

For each strength level, use a nested if to check if it's a bad password:

- **If Weak:**
 - If also a bad password: Print "Too short AND commonly used!"
 - If not a bad password: Print "Too short - use at least 6 characters"
- **If Medium:**
 - If also a bad password: Print "Common password detected - choose something unique"
 - If not a bad password: Print "Decent length - good job!"
- **If Strong:**
 - If also a bad password: Print "Long but predictable - avoid common passwords"
 - If not a bad password: Print "Excellent password!"

Sample Output 1 (weak and bad):

```
Enter password: admin
Password strength: WEAK
Too short AND commonly used!
```

Sample Output 2 (medium and good):

```
Enter password: mydog123
Password strength: MEDIUM
Decent length - good job!
```

Sample Output 3 (medium and bad):

```
Enter password: password
Password strength: MEDIUM
Common password detected - choose something unique
```

Sample Output 4 (strong and good):

```
Enter password: passwordpassword
Password strength: STRONG
Excellent password!
```

Implementation Hint: Structure your code with nested if statements inside each strength level check. Each outer if/elif checks the length, and the inner if checks whether it's a bad password.

8.2 Challenge 2: Smart Allowance Calculator (5 points)

Open [week8_ex8_2.py](#) and create a smart allowance calculator that calculates a student's weekly allowance based on age, chores, grades, and savings options. Follow the requirements below:

Problem Description

Create a program that calculates a student's weekly allowance based on multiple factors:

Base Allowance by Age:

- Under 10 years: Rs. 100
- 10-14 years: Rs. 150
- 15+ years: Rs. 200

Additional Factors:

- **Chores:** Add Rs. 20 for each chore completed
- **Grades:**
 - Grade A: Bonus (Rs. 30 for under 12 years old, Rs. 50 for 12+ years old)
 - Grade F: Penalty (Rs. 40 standard, or Rs. 20 if 3+ chores done)
- **Savings:** Option to save 20% (only if allowance \geq Rs. 100). For example, if allowance is Rs. 150, saving would be Rs. 30 and you would receive Rs. 120 in cash.

Technical Requirements:

- Must use `elif` chains for age groups
- Must include nested conditions for grade adjustments
- Must use boolean operators (`and`, `or`) for complex logic
- Display appropriate messages based on final allowance amount

Sample Output 1:

```
Enter your age: 12
How many chores did you complete? 5
What was your grade? (A/B/C/D/F): B
Do you want to save 20%? (yes/no): yes

Base allowance (10-14 yrs): Rs. 150
Chore bonus (5 x 20): Rs. 100
```

Total allowance: Rs. 250
 Saving 20%: Rs. 50
 Amount you get: Rs. 200

=====
 EXCELLENT WEEK! Great job!

Sample Output 2:

Enter your age: 16
 How many chores did you complete? 1
 What was your grade? (A/B/C/D/F): F
 Do you want to save 20%? (yes/no): no

Base allowance (15+): Rs. 200
 Chore bonus (1 x 20): Rs. 20
 Grade F penalty: -Rs. 40

Total allowance: Rs. 180
 Amount you get: Rs. 180

=====
 Keep trying! You can do better next week!

Submission Checklist

Before submitting, make sure you have:

- Completed all required sections
- Saved each code exercise in its own file
- Tested each code file individually to ensure it runs without errors
- Written your name at the top of this paper
- Answered all reflection questions
- Attempted bonus section (optional)
- Compressed all your code files into a single zip file named [week8_hw.zip](#)

Time Tracking:

How long did this homework take you? _____ hours

Parent/Guardian Signature: _____

(Confirming student completed work independently)

Excellent work mastering complex decisions!

Next week: Introduction to for loops