



## Homework Assignment

Week 5: Expressions and Operators

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: Operator Investigation (15 points)

*Time estimate: 20 minutes*

### 1.1 Exercise 1.1: Expression Evaluator (8 points)

Evaluate each expression step by step. Show your work using the order of operations!

Expression	Show Your Work	Result
$15 + 3 * 2$		
$(15 + 3) * 2$		
$20 / 4 + 3 ** 2$		
$100 - 25 * 2 + 10$		

### 1.2 Exercise 1.2: String Operations (7 points)

What happens when we use operators with strings? Fill in the results:

Expression	Result
<code>"Python" + "Fun"</code>	
<code>"Ha" * 3</code>	
<code>"2" + "3"</code>	
<code>len("Pakistan") * 2</code>	
<code>"Score: " + str(95)</code>	

## 2 Part 2: Expression Builder (20 points)

Time estimate: 25 minutes

### 2.1 Exercise 2.1: Shopping Calculator (10 points)

Open up file named `week5_ex2.1.py` and complete this program that calculates shopping discounts and tax:

```

1  # Smart Shopping Calculator
2  print("=== Eid Shopping Calculator ===")
3
4  # Original prices (in rupees)
5  shirt_price = 1200
6  shoes_price = 2500
7  scarf_price = 800
8
9  # Calculate subtotal
10 subtotal = ----- # Add all prices
11
12 # Apply 20% Eid discount
13 discount_amount = ----- # Calculate 20% of subtotal
14 after_discount = ----- # Subtract discount from subtotal
15
16 # Add 17% sales tax on discounted price
17 tax_amount = ----- # Calculate 17% of after_discount
18 final_total = ----- # Add tax to after_discount
19
20 # Display results with proper formatting
21 print(f"Subtotal: Rs. {-----}")
22 print(f"Eid Discount (20%): Rs. {-----}")
23 print(f"After Discount: Rs. {-----}")
24 print(f"Sales Tax (17%): Rs. {-----}")
25 print(f"Final Total: Rs. {-----}")
26
27 # Bonus calculation: If paying in 3 installments
28 monthly_payment = ----- # Divide final_total by 3
29 print(f"\nMonthly installment: Rs. {-----}")

```

Save this code in `week5_ex2.1.py`

### 2.2 Exercise 2.2: Sports Score Calculator (10 points)

#### Scenario: Cricket Match Statistics

In cricket, a batsman's strike rate shows how fast they score runs. It's calculated as:  $(\text{runs scored} / \text{balls faced}) \times 100$ . Also, the batting average is total runs divided by number of times out.

Open the file named `week5_ex2.2.py` and complete this program to calculate a cricket player's statistics:

```

1  # Cricket Statistics Calculator
2  player_name = "Babar Azam"
3
4  # Match 1 statistics
5  match1_runs = 85
6  match1_balls = 62
7
8  # Match 2 statistics
9  match2_runs = 120

```

```

10 match2_balls = 98
11
12 # Match 3 statistics
13 match3_runs = 45
14 match3_balls = 38
15
16 # Calculate totals
17 total_runs = _____ # Add all runs
18 total_balls = _____ # Add all balls
19 times_out = 2 # Given: out in 2 matches
20
21 # Calculate statistics
22 strike_rate = _____ # Use formula: (total_runs/total_balls)*100
23 batting_average = _____ # total_runs / times_out
24
25 # Calculate boundaries percentage
26 # Given: 15 fours and 8 sixes across all matches
27 boundary_runs = _____ # (15 * 4) + (8 * 6)
28 boundary_percentage = _____ # (boundary_runs/total_runs)*100
29
30 # Display results
31 print(f"=== {player_name} Statistics ===")
32 print(f"Total Runs: {_____}")
33 print(f"Strike Rate: {_____}")
34 print(f"Batting Average: {_____}")
35 print(f"Runs from Boundaries: {_____}%")

```

Save this code in `week5_ex2_2.py`

### 3 Part 3: Code Tracing Challenge (15 points)

Time estimate: 20 minutes

#### 3.1 Exercise 3.1: Order of Operations Trace (8 points)

Without running this code, trace through this program step by step:

```

1 a = 10 # Line 1
2 b = 5 # Line 2
3 c = 2 # Line 3
4 result1 = a + b * c # Line 4
5 result2 = (a + b) * c # Line 5
6 result3 = a ** c - b # Line 6
7 result4 = a / b + c * 3 # Line 7
8 final = result1 + result2 // 10 # Line 8
9 print(final) # Line 9

```

Fill in the value of each variable after each line executes. Put a cross (x) if the variable is not defined yet.

Line	a	b	c	result1	result2	result3	result4	final
3		x	x	x	x	x	x	x
4			x	x	x	x	x	x
5								
6								
7								
8								

What will the program print? \_\_\_\_\_

### 3.2 Exercise 3.2: Mixed Operations (7 points)

Without running the code, calculate what will each of these expressions evaluate to?:

```

1 x = 12
2 y = 5
3 z = 3
4
5 expr1 = x // y + z           # -----
6
7 expr2 = x % y * z           # -----
8
9 expr3 = x / y * z           # -----
10
11 expr4 = (x + y) % (z + 2)   # -----

```

## 4 Part 4: Boolean Logic Practice (20 points)

*Time estimate: 25 minutes*

### 4.1 Exercise 4.1: Comparison Detective (10 points)

Complete the table with True or False:

x	y	Expression	Result	Points
10	5	x > y		1 pt
10	10	x == y		1 pt
7	9	x != y		1 pt
15	20	x <= y		1 pt
"cat"	"dog"	x < y		2 pts
25	5	x % y == 0		2 pts
8	3	x // y >= 2		2 pts

### 4.2 Exercise 4.2: Logic Gates (10 points)

Open up the file named [week5\\_ex4.2.py](#) and complete this program that checks multiple conditions:

```

1  # School Event Eligibility Checker
2  age = 13
3  grade = 7
4  permission_slip = True
5  fee_paid = False
6
7  # Basic eligibility
8  old_enough = age >= 12           # Already done for you
9  correct_grade = -----         # Check if grade is 7 or 8
10
11 # Financial check
12 has_paid = -----              # Check if fee_paid is True
13 has_discount = age <= 13        # Younger students get discount
14
15 # Combined checks using AND
16 basic_eligible = ----- and ----- # old_enough AND correct_grade
17 can_go = ----- and -----      # basic_eligible AND
    permission_slip
18
19 # Combined checks using OR
20 payment_ok = ----- or -----   # has_paid OR has_discount
21
22 # Final eligibility using NOT
23 not_blocked = not -----        # Use NOT with some condition
24 fully_eligible = can_go and payment_ok
25
26 # Display results
27 print(f"Age check passed: {old_enough}")
28 print(f"Grade check passed: {correct_grade}")
29 print(f"Basic eligibility: {basic_eligible}")
30 print(f"Can attend: {can_go}")
31 print(f"Payment OK: {payment_ok}")
32 print(f"Final eligibility: {fully_eligible}")

```

Save this code in `week5_ex4_2.py`

## 5 Part 5: Program Development (25 points)

Time estimate: 30 minutes

Write a complete program in the given file `week5_ex5.py` that calculates a student's grade with weighted categories:

### Program Requirements

Your program should:

- Activity marks for Homework, Quizzes, Midterm, and Final Exam, each out of 100.
- Use these weights: Homework (30%), Quizzes (20%), Midterm (25%), Final (25%)
- Calculate the final semester score by adding weighted scores

### Example Run:

```

=== Grade Calculator ===
Enter homework average (0-100): 85
Enter quiz average (0-100): 92
Enter midterm score (0-100): 78
Enter final exam score (0-100): 88

```

**Grade Breakdown:**

Homework (30%):  $85.0 * 0.30 = 25.5$  points

Quizzes (20%):  $92.0 * 0.20 = 18.4$  points

Midterm (25%):  $78.0 * 0.25 = 19.5$  points

Final (25%):  $88.0 * 0.25 = 22.0$  points

Final Grade: 85.4%

**Testing Checklist:**

- Program runs without errors
- Correctly calculates weighted average
- Handles edge cases (100, 0, etc.)
- Output is well-formatted
- Uses expressions and operators correctly

**6 Part 6: Reflection Questions (5 points)**

*Time estimate: 10 minutes*

Answer these questions in complete sentences:

1. Explain why Python follows order of operations (BODMAS/PEMDAS). What would happen if it just went left to right?

---



---



---

2. When would you use integer division (//) instead of regular division (/)? Give a real-world example.

---



---



---

3. Which was harder to understand: arithmetic operators or boolean operators? Why?

---



---



---

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

*Only attempt if you've finished everything else!*

### 7.1 Challenge 1: Expression Puzzle (5 points)

Using only the numbers 2, 3, 4, 5 (each exactly once) and any operators, create expressions that equal:

- 24: \_\_\_\_\_
- 1: \_\_\_\_\_
- 100: \_\_\_\_\_

### 7.2 Challenge 2: Boolean Chain (5 points)

Write a single boolean expression (one line) that checks if a number is: - Between 10 and 100 (inclusive)  
- Even - Not divisible by 3

```

1 number = 44
2 is_special = _____
3 print(f"{number} is special: {is_special}")
    
```

## Submission Checklist

Before submitting, make sure you have:

- Completed all required sections
- Tested each code file individually to ensure it runs without errors
- Answered all reflection questions
- Attempted bonus section (optional)
- Compressed your code files into a single zip file named `week5_hw.zip`

### Time Tracking:

How long did this homework take you? \_\_\_\_\_ hours

**Parent/Guardian Signature:** \_\_\_\_\_

*(Confirming student completed work independently)*

**Great job mastering Python’s mathematical power!**

*Next week: Planning with Pseudocode and Flowcharts*